MINERALS

ECONOMIC & REGIONAL DEVELOPMENT AGREEMENT

Final Report

TASK 4 - GEOTECHNICAL DATABASE
FOR GROUND CONTROL

NUMERICAL MODELLING

by

INCO Limited, Thompson Division
Thompson, Manitoba

March 1990

Canada

Manitoba

MG-13097A

CANADA - MANITOBA

MINERAL DEVELOPMENT AGREEMENT

RESEARCH & DEVELOPMENT IN MANITOBA MINES

Work on the project was conducted under the auspices of Canada Centre for Mineral and Energy Technology, Energy, Mines and Resources Canada under the Canada/Manitoba Mineral Development Agreement.

DSS File NO. 14SQ.23440-4-9147-4
DSS Contract Serial No. OSQ84.00404
CANMET Project No. 4-9147-4
CANMET Scientific Authority: Y.S. Yu

Final Report

TASK 4 - GEOTECHNICAL DATABASE FOR GROUND CONTROL

NUMERICAL MODELLING

by

INCO Limited, Thompson Division
Thompson, Manitoba

March 1990

114 pp

MICROMEDIA

90-05661/2

CANADA - MANITOBA

MINERAL DEVELOPMENT AGREEMENT

RESEARCH & DEVELOPMENT IN MANITOBA MINES

GEOTECHNICAL DATABASE FOR GROUND CONTROL

NUMERICAL MODELLING

This has been prepared as part of the

TASK 4
Geotechnical Database for Ground Control
of the
Canada/Manitoba Mineral Development Agreement

by

J.A.C. (Tony) Diering
Steffen, Robertson, & Kirsten (B.C.) Inc.
Sub-contracting to
Inco Limited, Thompson Division, Manitoba

DSS File No.: 14SQ.23440-4-9147-4
CANMET Project No.: 4-9147-4
CANMET Scientific Authority: Y.S. Yu

CANMET TASK 4

NUMERICAL MODELLING

FINAL REPORT

STEFFEN ROBERTSON AND KIRSTEN INC.

REPORT NO. 54703.4

JANUARY 1990

ABSTRACT

This report describes the development and testing of four computer programs for numerical stress analysis of underground excavations. The programs are QUAD, a two-dimensional finite element analysis package, BEAP, BEAP-DD and BEAP-M, which are three-dimensional boundary element analysis packages. The programs represent further developments of earlier programs (QUAD and MBEM) developed by Steffen, Robertson and Kirsten Consulting Engineers.

It was intended to improve the usability and capability of existing software, in particular to enable multi-step analyses, improved modelling of open-pit/underground interactions and non-linear behavior to be performed.

This study has taken place over 5 years, and although it is felt that the overall objectives as specified 5 years ago have completed, it is evident that certain aspects have been dealt with more thoroughly than others. It is felt that the programs developed represent a positive addition to the numerical modelling toolbox of the rock mechanics practitioner in Manitoba and elsewhere.

Five additional years of experience in the use of three-dimensional stress analysis programs have demonstrated, however, that suitable training and experience are necessary before these programs may be effectively used and this is perceived as the biggest single drawback preventing efficient use of this software in the future.

SOMMAIRE

Ce rapport décrit le développement et l'expérimentation de quatre programmes informatiques d'analyse numérique des contraintes en excavation souterraine. Les programmes sont QUAD, un programme bi-dimensionnel d'éléments finis et BEAP, BEAP-DD et BEAP-M, des programmes tri-dimensionnels d'éléments frontières. Ces programmes représentent de nouveaux développements sur les anciens programmes (QUAD et MBEM) développés par Steffen, Robertson and Kirsten Consulting Engineers.

Il était projeté d'améliorer l'application et la capacité des logiciels déjà existants, en particulier de rendre possible l'analyse par étapes multiples, d'améliorer la modélisation de l'interactions fosse/souterrain et d'implanter les comportements non-linéaires de la roche.

Cette étude s'est déroulée sur une période de 5 ans, et bien que tous les objectifs ont été complétés Il est évident que certains aspects ont été abordés avec plus d'instance que d'autres. À ce jour, les programmes développés représentent un ajout positif aux outils de la modélisation numérique de la mécanique des roches appliquées au Manitoba et partout ailleurs.

Cette expérience additionnelle de cinq ans sur l'utilisation des programmes d'analyse tri-dimensionnel des contraintes ont, cependant, démontré qu'un entraînement et qu'une certaine expérience sont nécessaires avant d'utiliser ces programmes d'une façon efficace. Ceci est perçu comme le plus gros obstacle à surmonter avant d'utiliser efficacement les logiciels dans le futur.

TABLE OF CONTENTS

			Page
	ABSTRACT		
1.0	**INTRODUCTION**		1
2.0	**TECHNICAL OVERVIEW**		5
	2.1	QUAD	5
		2.1.1 General features	5
		2.1.2 Digitizer facilities	6
		2.1.3 Plotting facilities	6
		2.1.4 Keyword control	8
		2.1.5 New database structure	12
	2.2	BEAP	13
		2.2.1 Numerical formulation	13
		2.2.2 Evaluation of Matrix Coefficients	15
		2.2.3 Assembly and solution of equations	15
		2.2.4 Multiple subregion problems	17
		2.2.5 Computational aspects	17
	2.3	BEAP-DD	20
		2.3.1 Numerical formulation	20
		2.3.2 Description of lumps	20
		2.3.3 Influence coefficients and lumping	22
		2.3.4 Computational aspects	26
	2.4	BEAP-M	32
		2.4.1 Numerical formulation	32
		2.4.2 Influence coefficients and lumping	34
		2.4.3 Equation solution	35
		2.4.4 Benchmarks	36
		2.4.5 Computational considerations	36

3.0	**TEST EXAMPLES**		**37**
	3.1	QUAD	37
		3.1.1 Distributed joint elements	37
		3.1.2 Circular tunnel	39
	3.2	BEAP	44
		3.2.1 Two-bonded cubes	44
		3.2.2 Tunnel with square cross section	44
	3.3	BEAP-DD	51
		3.3.1 Small folded seam	51
		3.3.2 Circular excavation	58
		3.3.3 Shallow reef example	61
	3.4	BEAP-M	75
		3.4.1 Solid cube containing a seam	75
4.0	**PRACTICAL EXAMPLES**		**81**
	4.1	Underground excavations - QUAD	81
	4.2	Five interacting excavations BEAP	81
	4.3	Eight excavation problem -- BEAP	88
	4.4	Shallow rcef example -- BEAP-M	91
	4.5	Open pit/underground example	91
5.0	**DISCUSSION**		**98**
	5.1	ORIGINAL OBJECTIVES	98
		5.1.1 Mining sequence	99
		5.1.2 Structural discontinuities (faults and joints)	99
		5.1.3 Surface/ underground interaction	99
		5.1.4 Multiple or folded seams	99
		5.1.5 Improved windowing procedures	100
		5.1.6 Post failure material	100
		5.1.7 Failure criteria for post processing	100
		5.1.8 Energy Release	100
		5.1.9 User friendly/restart	100
		5.1.10 Fortran 77	101
		5.1.11 User documentation	101
		5.1.12 Pre-processing	101
		5.1.13 Post processing	101
	5.2	OTHER FEATURES	102
6.0	**CONCLUSIONS**		103
7.0	**REFERENCES**		104

LIST OF FIGURES

1.1	Relationship between Tasks 1,4 and 5.	2
1.2	Geometry for typical BEAP-M problem	4
2.1.2.1	Digitized mesh (showing some errors)	7
2.1.3.1	Geometry and material properties for non-linear test problem (QUAD)	9
2.1.3.2	Non-linear test problem - displacement vectors	10
2.2.1.1	Typical types of boundary element.	14
2.2.1.2	Node locations for non-conforming quadratic elements	16
2.2.4.1	Treatment of interface elements during iterative solution process	18
2.3.2.1	Local and intrinsic co-ordinates for a lump, also showing element numbering	21
2.3.2.2	Global (G) and local coordinates for emitter (E) and receiver (R) lumps	23
2.3.3.1	Different types of influence coefficients between two lumps	25
2.3.3.2	Results for two lump DD test problem	27
2.3.4.1	BEAP-DD Input flow diagram	28
2.3.4.2	BEAP-DD Number crunching flow diagram	29
2.3.4.3	BEAP-DD Post processing flow diagram	30
2.4.1.1	Different types of element interaction required for BEAP-M	33
3.1.1.1	Geometry and material properties for joint example	38
3.1.2.1	Circular tunnel test problem : linear vs non-linear	41
3.1.2.2	Circular tunnel test problem : QUAD vs analytic	42
3.1.2.3	Circular tunnel test problem : Fine mesh solution	43
3.2.1	Geometry for unit cube test problem	45
3.2.2	Geometry for shperical cavity test problem	46
3.2.3	Geometry for Cube excavation test problem	47
3.2.1.1	Geometry for two bonded cubes	48
3.2.1.2	Results for two bonded cubes	49
3.2.2.1	Comparisons of horizontal displacements for tunnel problem	50
3.3.1.1	Geometry for folded seam test problem	52
3.3.1.2	Cross section through two intersecting reefs (folded seam) test problem.	53
3.3.1.3	Closures and rides for reefs intersecting at 135 degrees	54
3.3.1.4	Closures and rides for reefs intersecting at 90 degrees	55
3.3.1.5	Closures and rides for reefs intersecting at 60 degrees	56
3.3.1.6	Closures and rides for reefs intersecting at 45 degrees	57
3.3.1.7	Comparison of influence coefficients of LE = 2, NE = 4 on LR = 1, NR = 3	59
3.3.1.8	Closure and ride along a section of two intersecting reefs problem	60
3.3.2.1	Closures for a circular excavation	62
3.3.3.1	Dipping reef close to earth's surface	64
3.3.3.2	Dipping reef close to earth's surface - surface closures	65
3.3.3.3	Dipping reef close to earth's surface - surface rides	66
3.3.3.4	Dipping reef close to earth's surface - reef closures	67
3.3.3.5	Dipping reef close to earth's surface - reef rides down dip	68
3.3.3.6	Dipping reef without earth's surface - closures	69
3.3.3.7	Dipping reef without earth's surface - rides down dip	70
3.3.3.8	Dipping reef closures with and without surface	71
3.3.3.9	Dipping reef rides with and without surface	72
3.3.3.10	Dipping reef with and without earth's surface - comparison of 2D and 3D 0 - closures	73
3.3.3.11	Dipping reef with and without earth's surface - comparison of 2D and 3D - rides	74

3.4.1.1	Geometry for solid cube test problem - BEAP-M	76
3.4.1.2	Vertical displacements - Seam unmined	77
3.4.1.3	Vertical displacements - Seam totally closed	78
3.4.1.4	Tractions/stresses - Seam unmined	79
3.4.1.5	Tractions/stresses - Seam totally closed	80
4.1.1	Finite element mesh for underground mining problem	82
4.2.1	Five excavation problem (looking down)	84
4.2.2	Five excavation problem (looking up)	85
4.2.3	Five excavation problem - location of benchmarks A and B	86
4.3.1	Eight excavation problem - BEAP mesh	89
4.3.2	Eight excavation problem - four mining steps.	90
4.4.1	Shallow reef example - geometry and closure	92
4.4.2	Shallow reef example - closures along a section	93
4.4.3	Shallow reef example - rides along a section	94
4.4.4	Shallow reef example - Selected surface stress results	95
4.4.5	Shallow reef example - Horizontal displacement vectors at surface.	96
4.5.1	Close up of crown pillar area.	97

1.0 INTRODUCTION

In 1985, several projects were initiated under the auspices of the Canada Centre for Mineral and Energy Technology, Energy, Mines and Resources Canada under various Canada/Manitoba Mineral Development Agreements. A number of these were to be conducted at the Manitoba Division of INCO at Thompson (figure 1.1). This study entitled 'TASK 4 - GEOTECHNICAL DATA BASE FOR GROUND CONTROL' was subdivided into two main parts, namely 'A Geomechanical database' and 'Numerical modelling'. This report is concerned solely with the numerical modelling aspects of Task 4. Other aspects are reported separately.

This report describes the work which been done, aspects which have been comprehensively covered, and aspects which have received less attention. The report itself is of a summary nature, since most of the work is contained within the actual source code developed and the user documentation for the various programs and the testing of the software.

The numerical modelling aspect of Task 4 has been broken down into the development of four programs, namely QUAD, BEAP, BEAP-DD and BEAP-M. The planned functionality of each of these programs is as follows:

QUAD	A two-dimensional finite element analysis for plain strain analyses of linear or non-linear non-homogeneous rock masses. There is no direct link between QUAD and the BEAP programs above. However, the use of QUAD is intended to be complimentary to the use of the BEAP programs.
BEAP	Three-dimensional boundary element stress analysis of open or massive underground excavations. Excavations are subdivided into irregular quadrilateral surface or boundary elements. These elements may also be used to model the earth's surface.
BEAP-DD	Three-dimensional displacement discontinuity element analysis of tabular underground excavations. Each element is assumed to be square, but different sized and oriented elements may be used in different locations within a given problem.
BEAP-M	A Mixed formulation combining most of the features available to the above programs.

Figure 1.1: Relationship between Tasks 1, 4 and 5

In summary, the functionality of the various programs as related to numerical stress analysis of mining rock mechanics problems is described in the table below:

Item	QUAD	BEAP	BEAP-DD	BEAP-M
2- or 3-dimensional	2	3	3	3
Non-linear	yes	no	some	some
Geology	good	poor	-	-
Open excavations	good	good	-	-
Tabular excavations	fair	poor	good	good
Mixed open/tabular	yes	some	no	yes
Multi-step	yes	no	yes	yes
Open pit/underground	yes	yes	no	yes
Typical problem size (degrees of freedom)	2000	4000	10000	15000
Hardware (IBM-PC, mainframe or Both)	IBM-PC	IBM-PC	Both	IBM-PC

Most of above modelling aspects are available within other programs available from various countries. However, at the start of this project (and probably at the end of the project as well), programs for modelling mixed open/ tabular type excavations three-dimensionally (see figure 1.2) were not generally available, with the exception of program MBEM, which has formed the basis for the development of the BEAP suite of programs.

This report contains a description of each program, including descriptions of the construction and solution of the governing equations and other computational considerations. Numerous test and other examples are also described briefly.

It should be recognized that most of the development effort has been put into the BEAP suite of programs, with less than 25% of the effort being directed towards QUAD, the two-dimensional finite element code.

It was stated that all software developed should be in ANSII standard FORTRAN. This objective has been largely complied with, as is evidenced by the fact that all programs will execute on IBM-PCs, whilst BEAP has been installed on a VAX mini-computer, and BEAP-DD has been installed on the IBM mainframe computer at Thompson. The preferred and recommended hardware for the software is the IBM-PC compatible hardware.

Finally, under the concluding remarks, some thoughts on the overall success/failure of this project are presented.

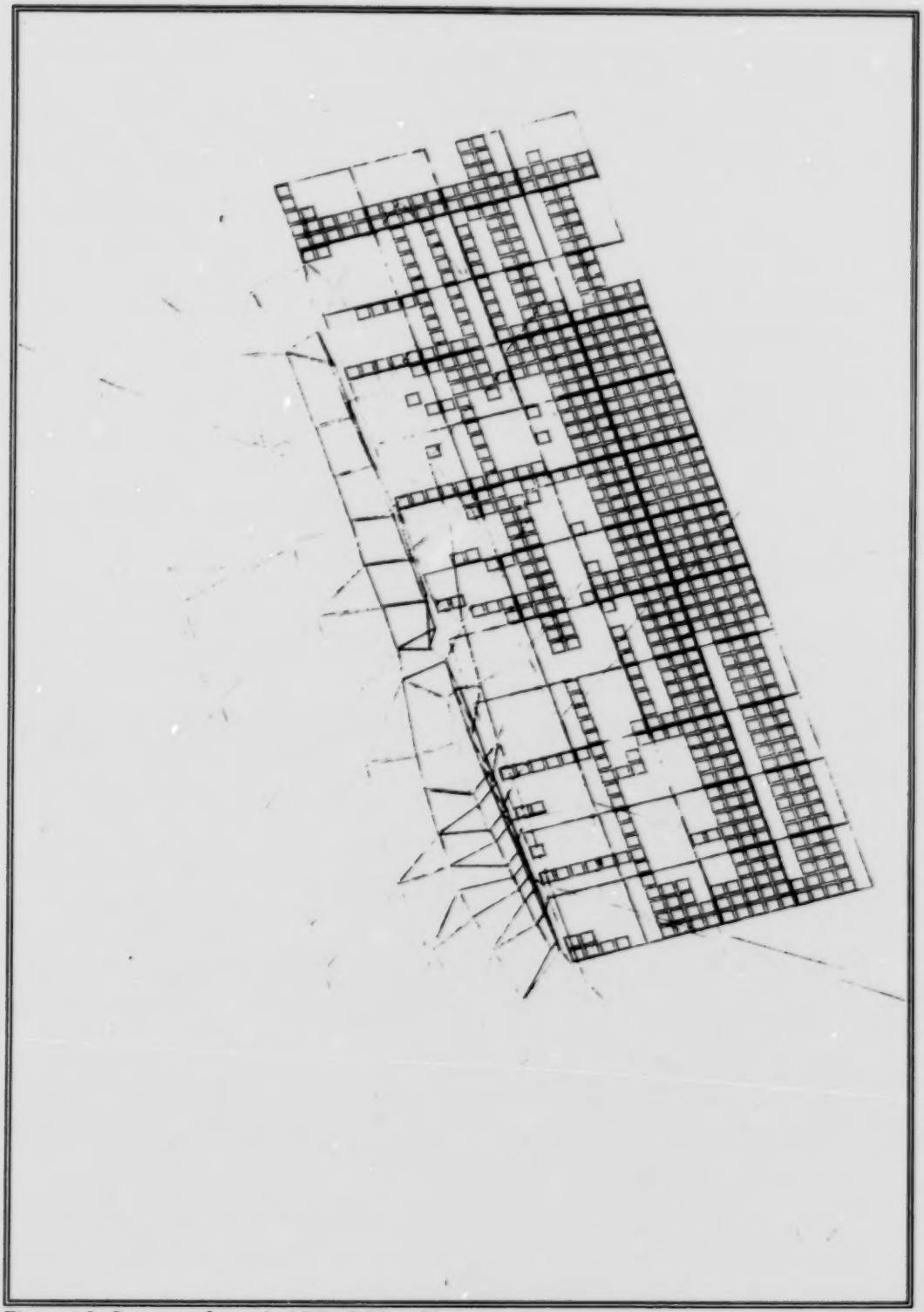


Figure 1.2: Geometry for typical BEAP-M problem

2.0 TECHNICAL OVERVIEW

2.1 QUAD

2.1.1 General features

Program QUAD was originally developed as a two-dimensional finite element code by Steffen, Robertson and Kirsten during the early 1980's. It has gradually evolved and was installed on the IBM mainframe computer at INCO Thompson in 1983. However, as the program evolved, so its internal program structure became progressively more untidy and less user friendly. One of the objectives for this project was to restructure the program internally, so as to facilitate further development and the extended specification of Task 4. Particular development directions were improved non-linear modelling and multi step capability.

The process of restructuring required that numerous separate smaller programs be combined and unified into one system. Work was initially carried out on a Data General computer. During the year, QUAD was transferred onto an IBM-PC/AT and subsequent development was carried out on this computer.

One of the main objectives was to implement a 'unified' database for use by all program modules. Previously, (for example the version of QUAD installed at Thompson in 1983) numerous independent programs were run, each with its own input and output datafiles. Management of these various datafiles was tedious and prone to errors. With the new implementation, it was intended to have one database for each problem which contained all information in a manner which could easily accessed for viewing or alteration by the user. Originally, an attempt was made to enable all data to be entered into the system interactively. However, it subsequently proved to be more efficient to make use of datafiles pre-prepared by the user. Program modules were built into QUAD to assist the user in preparing these files (for example the interactive digitizing and mesh double up facilities).

As regards the non-linear modelling capabilities of QUAD, various features have been implemented. These work well, but it should be noted the non-linear modelling capability is still restricted by the limits imposed by the basic formulation in use here, namely linear isoparametric quadratic elements for plain strain. For example, grossly non-linear geometric behavior cannot be modelled.

Joint elements (distributed or ubiquitous) had been programmed into QUAD for some time. During the course of the project, the joint elements have been modified, improved and better tested. In addition, a dilatancy parameter was built into the formulation.

The Mohr-Coulomb failure criterion was extended to allow for variable rock dilatancy. This is a particularly useful feature for brittle rock masses which are known to exhibit dilatant behavior during failure.

In the five years since this project was initiated, numerous finite element packages have been developed and marketed for various disciplines. However, during every phase of the development, considerable care was taken to ensure that QUAD remained a package designed primarily for application to mining rock mechanics problems. It is felt that QUAD is still a current and useful package for this class of problems

2.1.2 Digitizer facilities

There are numerous examples reported in the literature of mesh generators for all kinds of finite element packages. A fundamental disadvantage of the mesh generator approach (as they relate to geological and mining problems) is that as the combination of geology and mine excavations becomes progressively more complex and irregular, so the mesh generators become progressively more cumbersome or difficult to run. In other words, mesh generators work well for regular meshes, but less well for an irregular mesh. Another property of mesh generators in general is that they often waste elements (ie they generate more elements in some areas than are really needed).

For this reason, and also through experience, it was felt advantageous to include the use of a digitizer for mesh input. The big advantage of using a digitizer is that the user draws up a mesh as he wants it, so that it best models his combination of geology and mining geometry. Having drawn a mesh up by hand (usually not too tedious a task), a typical mesh of 500 to 1000 nodes can usually be digitized in an hour or two. Another advantage of a digitizer approach is that the user does not have to learn about various rules for mesh generation before he may digitize a mesh. A disadvantage, however, that a digitizer represents additional hardware which might not always be available.

The digitizing part of QUAD performs the following basic functions

- digitize nodes
- move nodes which have already been digitized
- digitize elements (5 points digitized per element - one for the element centre, and one for each of four nodes)
- digitize elements (1 point digitized per element - useful for digitizing regular parts of a mesh)
- digitize boundary condition codes

As an example, a sample mesh for a room and pillar system was drawn up. The mesh considers one quarter of one pillar and one quarter of the stope, the remainder being taken into account by symmetry. This mesh may be digitized in around half an hour using the digitizing facilities. A plot of the mesh, which also shows a few deliberate errors, is shown in figure 2.1.2.

2.1.3 Plotting facilities

The plotting process used by QUAD takes place in two stages. In the first (which is handled by QUAD), plot files are prepared for various items as selected by the user. The second stage consists of actually displaying these prepared files onto a graphics device. QUAD is primarily concerned with the preparation of various plot files, as this can be made to be relatively device independent.

Plotting modules (used to prepare these plot files) are as follows:

Prepare a mesh plot

Prepare boundary conditions for plotting

Prepare perimeters for plotting

Prepare element results for plotting

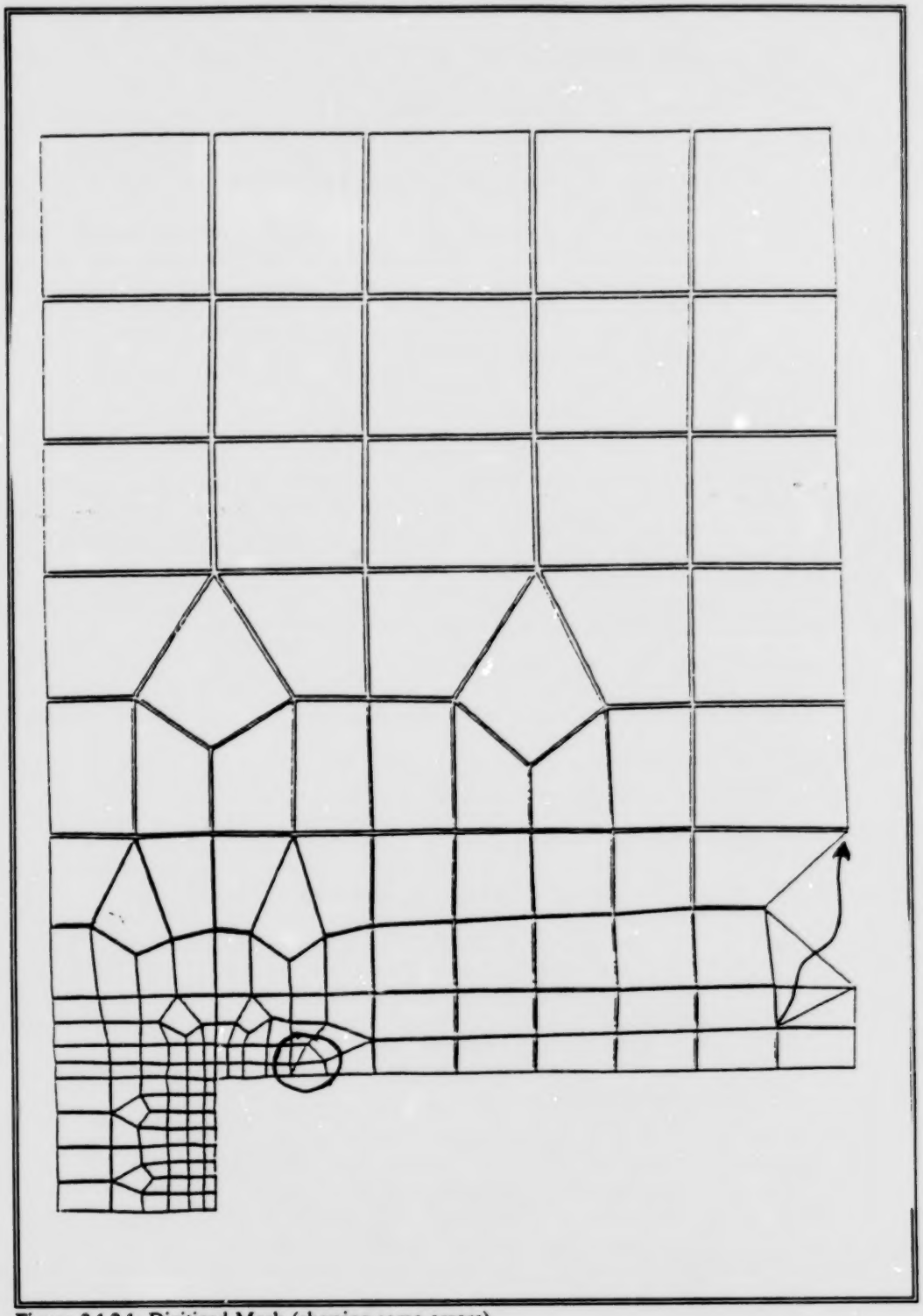


Figure 2.1.2.1: Digitized Mesh (showing some errors)

Prepare nodal results for plotting

Prepare a deformed mesh for plotting

Details of plotting capabilities are contained within the user documentation.

An example problem and an example plot of displacement results are shown in figures 2.1.3.1 and 2.1.3.2. This example was used some time ago to test the Mohr Coulomb failure criterion. The displacement vectors shown in figure 2.1.3.2 were of interest because they show the softer material (E = 10) trying to 'flow' around the stiffer material (E = 1000). As QUAD is incapable of modelling large displacements, it is not really modelling this flow. Rather, the model shows the 'sense' in which flow or movement is taking place.

2.1.4 Keyword control

It was originally intended that all input into the QUAD database for a given problem should be done interactively. However, experience has shown that for realistically sized problems, the preparation of one or several input datafiles is still preferable, particularly in view of the ever increasing and abundance of powerful text editors.

The keywords described below form the basis for input of data into the QUAD database. Detailed descriptions of selected keywords are included in the user documentation.

KEYWORD	Description
STOP	Stop execution
MENU	Get the interactive menu
START	Start a totally new run -- initialize everything
STIFF	Update the stiffness matrix (after changes to material types)
DATA	Change to a new data file (Filename entered from keyboard)
PERI	Read sequences of nodes defining perimeters
RUN	Start execution for current step
NODES	Start reading nodes
ELTS	Start reading elements
MATP	Start reading material properties
BC	Start reading boundary conditions at nodes
LIN	Set a linear run
EX	Define stresses around excavation perimeters

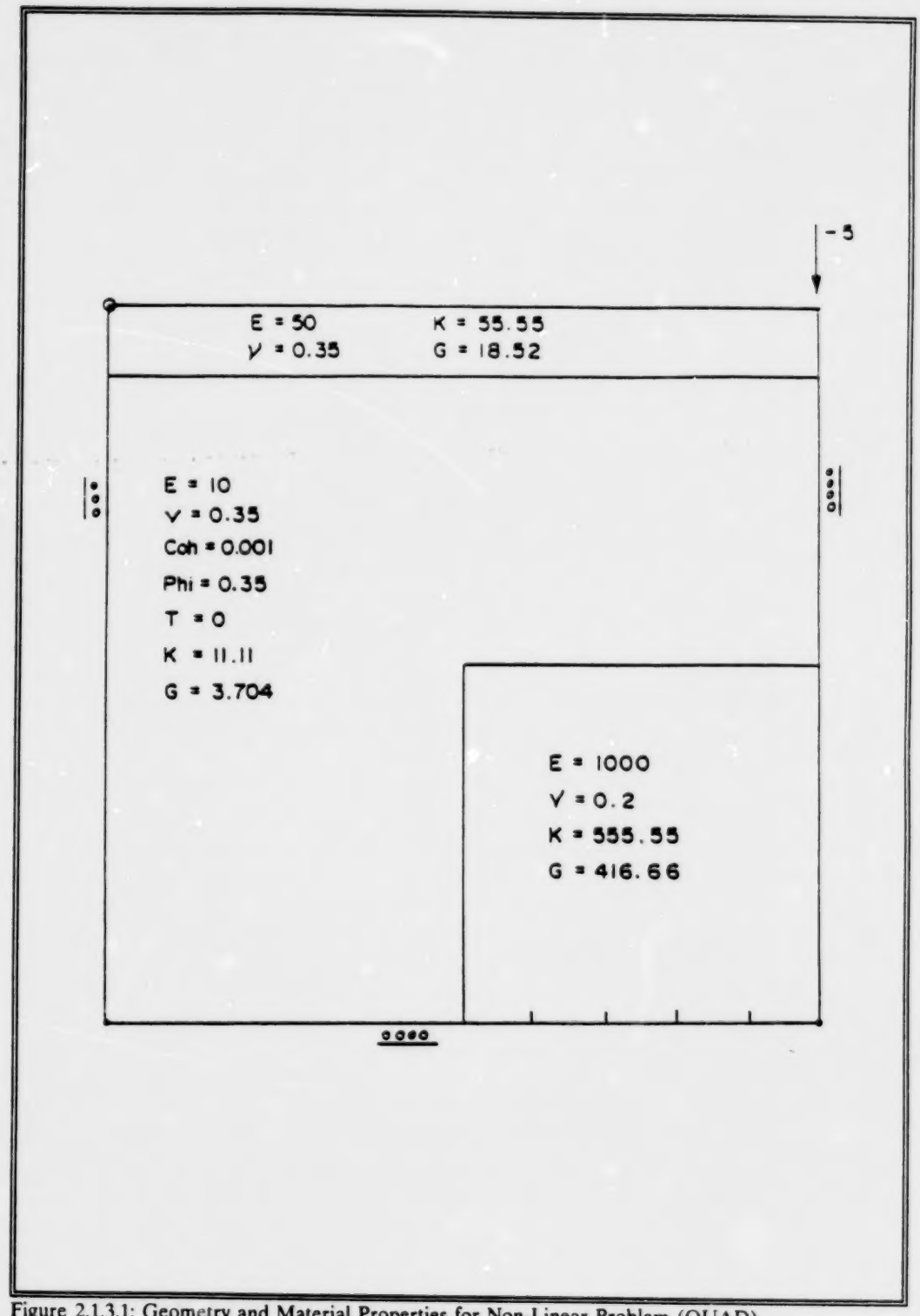


Figure 2.1.3.1: Geometry and Material Properties for Non-Linear Problem (QUAD)

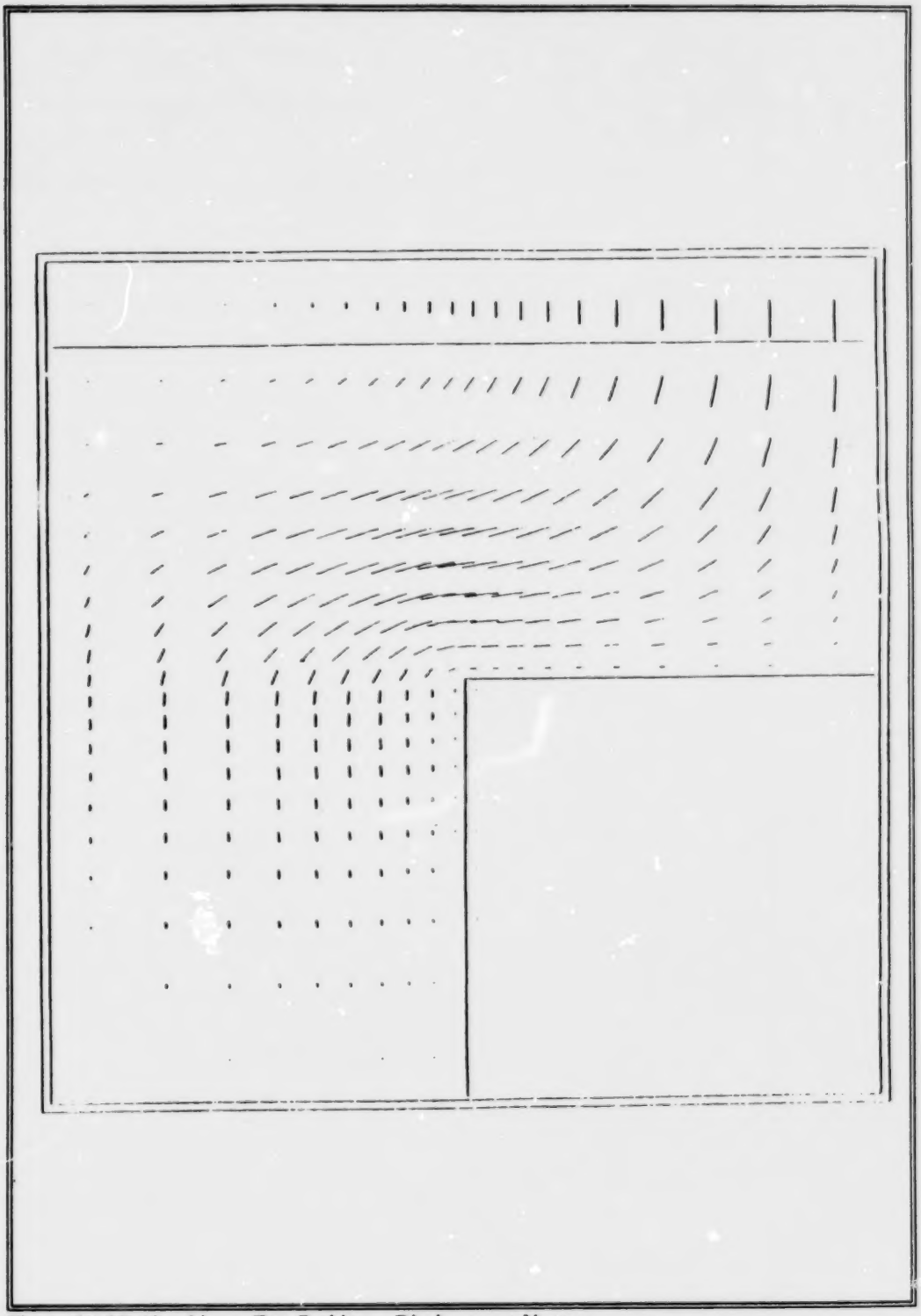


Figure 2.1.3.2: Non-Linear Test Problem - Displacement Vectors

NLIN	Set a non-linear run
STEP	Read in multi step information
INIT	Initialize stresses, displacements, strains and forces
ISSW	Define integer switches
SSW	Define real switches
PRIM	Read in primitive stress conditions
OUT1	Print out header information
OUT2	Print node/element geometry for easy re-input
OUT3	Print selected element geometry
OUT4	Print out material properties
OUT5	Print out nodal solutions
OUT6	Print out element solutions
OUT7	Print out perimeter information
MATEL	Specify material type for each element
BOXBC	Specify boundary conditions to all nodes inside a box
LINBC	Specify line load bc's, node by node
ZBC	Initialize all bc's to zero
PAIN	Plain stress analysis
PESS	Plain stress analysis
AXIS	Axi-symmetric analysis
GRAV	Add gravity load at specified elements
JOINT	Input joint properties
ENDBC	End boundary condition input and also set element properties
BUG1	Display system info (related to common block)
BUG2	Display system info (related to arrays)
VIRG	Allocate virgin stress conditions to selected elements
CONT	Use a contour to set primitive stresses at given surface

MESH	Prepare mesh for plotting
PLBC	Prepare boundary conditions for plotting
DOUBLE	Double up a mesh
DIGIT	Enter the digitizer menu
HEAD	Read in user heading/problem description
NONI	Input non-linear iteration parameters
DEFM	Deformed mesh plot preparation
ELTP	Prepare element results for plotting
NODP	Prepare nodal results for plotting
PERP	Prepare perimeters for plotting

2.1.5 New database structure

In this section, reasons for developing a new data base structure are described. An overview of the new structure is also presented for completeness.

Preparation of input for a finite element analysis is often a tedious process. A mesh has to be drawn up (or generated using a mesh generator). Thereafter, material properties have to be assigned to different groups of elements. Nodal boundary conditions have to be defined as well as load conditions. It is also usual to prepare various plots of the input data for data checking.

In the previous versions of QUAD, many of the above processes were carried out by different programs, each with its own input and output files. This resulted in numerous files being handled by the user with the resulting increased chances of data errors. The new database structure was aimed to alleviate this problem by having one unified database for mesh, boundary conditions and material properties. Each program module adds to or modifies the main database. One reason for splitting the database into several smaller files, is that when programs are run on an IBM-PC/AT, it is sometimes desirable to put some of the files onto a virtual disk or ramdrive. By doing this, the overall execution time of some problems can be speeded up considerably. Another reason for having the database stored on many files is that when several runs are carried out, it may not be necessary to backup all files for every run. For example, the global stiffness matrix file only changes when material properties of displacement boundary conditions are changed. In this manner, storage requirements for backups is reduced.

File names and descriptions are given in the following table. It should noted that the channel numbers are primarily for internal program usage and not of direct concern to users.

Number	Filename	Description
21	QUAD.COM	Main common block
22	QUAD.ELT	Element geometry file
23	QUAD.MAT	Material properties file
24	QUAD.ARR	Big arrays file
25	QUAD.STF	Global stiffness matrix file
26	QUAD.PER	Perimeters file
27	QUAD.EST	Element stiffness file
28	QUAD.MUL	Multi step control file (not used yet)
29	QUAD.NOD	Nodal stresses/strains (not used yet)
30	DIGITIZER	Channel for reading from digitizer
31	QUAD.PV	Plot file for vectors
32	QUAD.PG	Plot file for General items
33	QUAD.PS	Plot file for stresses
34	QUAD.PI	Plot file for integer numbers
35	QUAD.PN	Plot file for node numbers
36	QUAD.PE	Plot file for element numbers
37	QUAD.PM	Plot file for mesh
38	QUAD.PD	Plot file for deformed mesh
39	QUAD.PP	Plot file for perimeters
50	QUAD.MES	Message file

2.2 BEAP

2.2.1 Numerical formulation

The governing equations for three-dimensional elastostatics formulations are well developed and do not need to be repeated here. In their simplest form, they can be written as

$$Tu = Ut \qquad (1)$$

where T and U are matrices of coefficients derived from the fundamental displacement and traction (Kelvin) solutions for a point force in an infinite space, u is a vector of displacement components and t is a vector of traction components.

The T matrix includes the so called cij terms associated with the evaluation of the nearly singular integrals. Standard procedures are used to subdivide the surfaces of the mine excavations and the earth's surface into boundary elements. In this formulation, use is made of non-conforming quadratic elements. Figure 2.2.1.1 shows how these elements compare with other types of boundary element. Each boundary element is defined using from 3 to 9 nodes (referred to as geometric nodes). Four geometric nodes would be used to define a planar quadrilateral, for example.

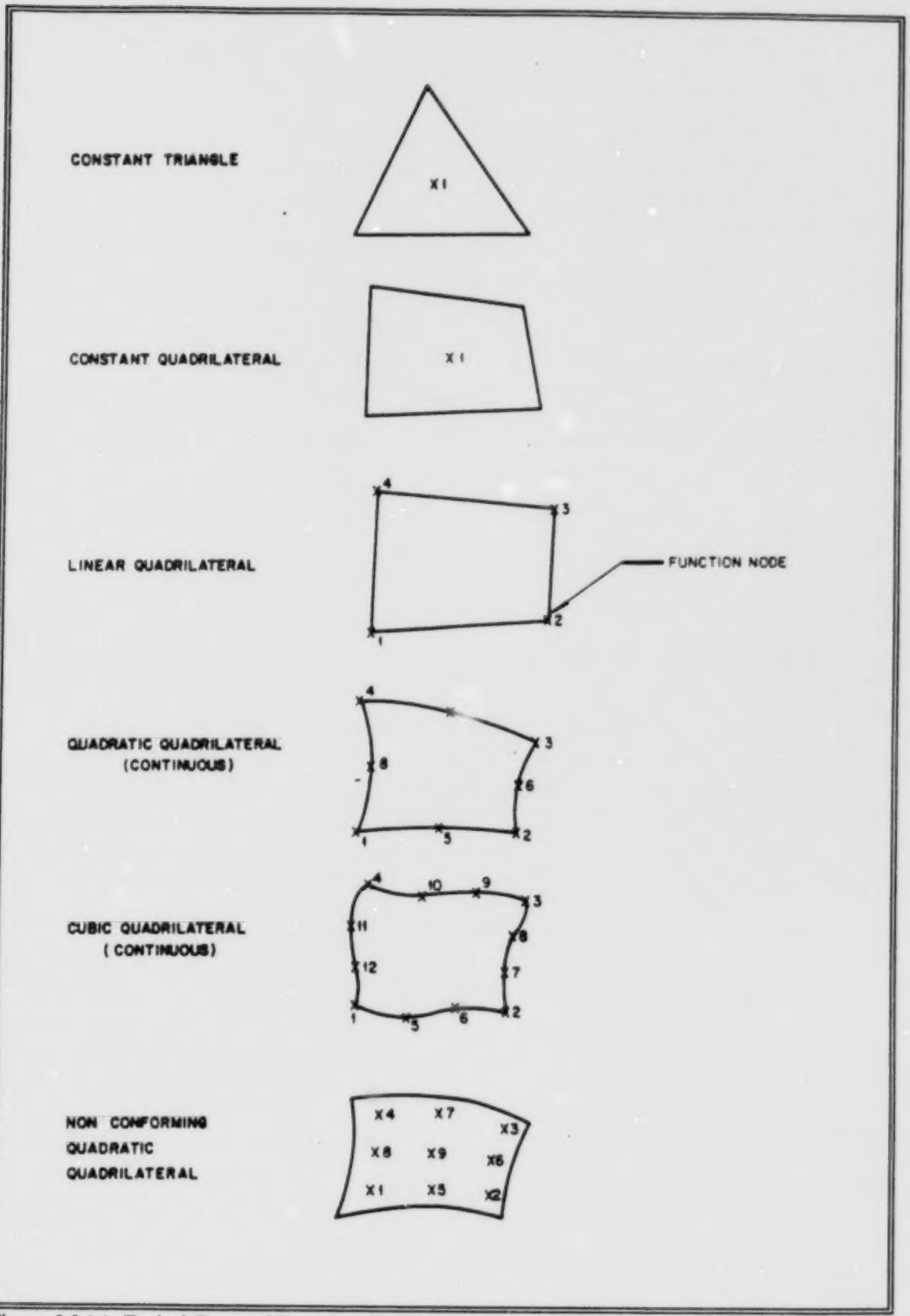


Figure 2.2.1.1: Typical Types of Boundary Elements

Using the element geometry as defined by the geometric nodes, nine internal "function" nodes for displacements and tractions may then be defined.

Quadratic Lagrangian shape functions are used to describe the displacement and traction variation over each element. As the function nodes are not located at the edges of elements, tractions and displacements may be discontinuous between elements. The function nodes have been located at local coordinates (intrinsic to each element) of +-0.8 or +-0.66667, with element edges being defined by local coordinates of +-1. (See figure 2.2.1.2)

Each element has 9 function nodes and hence 27 degrees of freedom. This holds even if triangular elements are used. Triangular elements are treated as degenerate quadrilaterals but, as the degeneracy is located at the edge of the element, the function nodes which are located within the element are not affected.

2.2.2 Evaluation of Matrix Coefficients

Evaluation of the integrals which define the matrix coefficients is carried out using the already well established method of Gaussian quadrature of variable order, depending upon the separation between the source (or receiver) node and the field (or emitter) element under consideration. If the receiver node is too close to the emitter element, then the emitter element is subdivided into subelements.

Special, but again well established, procedures are required to evaluate the diagonal matrix coefficients. However, since the function nodes are located within the borders of each element, the diagonal terms for the T matrix will usually have a value of 0.5. Different values are obtained when symmetry and/or repeatability are used.

The method of integration has been found to work effectively in practice, although more efficient methods have now been developed.

2.2.3 Assembly and solution of equations

The matrices T and U can be subdivided into submatrices Tmn and Umn. Tmn and Umn describe the influence of an emitter element n upon a receiver element m. As each element contains 9 nodes and 27 degrees of freedom, each submatrix will contain 27 x 27 = 729 coefficients. The size of these submatrices is very conveniently handled in a personal computer environment. However, if the number of elements is large, then the total disk space requirements can become prohibitively large. The design size for the formulation was at least 256 elements for a single subregion. Under these conditions, up to 382 Megabytes of disk storage could be required if coefficients are stored using double precision. As current personal computer (and even mini-computer) technology makes available disk space of from 20 to 140 Megabytes typically, a means of reducing disk storage requirements was required.

The decision was taken to use an iterative equation solver with lumping. Such techniques are already well established in the mining industry's use of displacement discontinuity elements where large systems of equations have been solved routinely. However, because iterative equation solvers perform very badly when beam bending effects are present, they have generally been unpopular in boundary element formulations designed for structural problems. The main advantages of the iterative solver implemented in the program BEAP are:-

- Use of single precision arithmetic

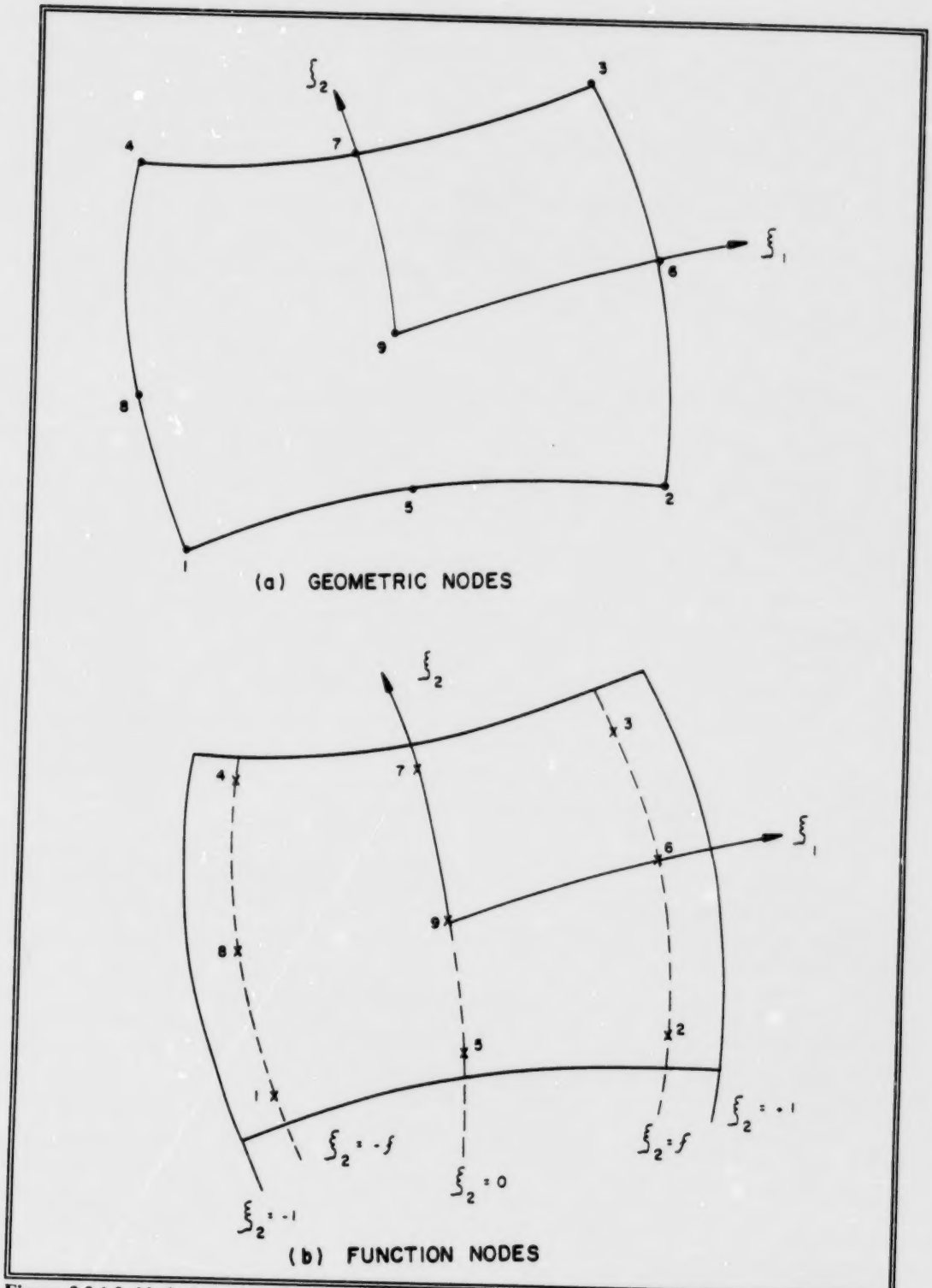


Figure 2.2.1.2: Node Locations for Non-conforming Quadratic Elements

- Use of lumping
- Reduction of solution time (as compared with elimination techniques) except where beam bending is present or for certain non-homogeneous problems
- Previous experience with the method as applied to mining problems
- Ease of programming

2.2.4 Multiple subregion problems

Other workers in the field of boundary elements have shown that problem size can be reduced in many cases by subdividing one large region into a number of subregions [5,6]. The ability to use subregions has been included in BEAP, but its use is discouraged except when the subregions have very different material properties. The reason for this is that most problems in mining involve either an infinite space or a half space. These problems do not lend themselves to a ready subdivision into subregions as is often the case with finite geometries.

When, however, use of subregions is necessary, there is a strong possibility that loss of numerical convergence of the iterative equation solver may occur[4,14]. With a constant element formulation, there were two problem areas, the first being the possible loss of (near) diagonal dominance of the matrix equations and the second being the inability of constant elements to model rigid body translation with rotation. The latter case arises, for example, when an attempt is made to model a stiff object which is completely enclosed by a soft surrounding subregion.

When quadratic non-conforming elements are used, however, rigid body translation with rotation can be modelled, and reasonable diagonal dominance of the matrix equations can be obtained as follows. At any subregion interface, elements are specified such that tractions are applied to the stiff subregion, and displacements to the softer subregion. This method has been shown to work reasonably well (See figure 2.2.4.1). However, more work in this field is still required.

2.2.5 Computational aspects

This section describes some aspects of the computer code for BEAP which are closely associated with the use of non-conforming elements. Undoubtedly, the biggest single advantage of using these elements has been that of a program which (on a Data General Computer) has required less than 64 kbytes of processing memory in total for both the program code and data. Accordingly, the process of "down-loading" the program onto an IBM-PC/AT with 640 kbytes of memory was relatively trivial.

It has already been stated that use of single precision and lumping results in a coefficient matrix (stored on disk) which is of manageable size. The main process whereby memory requirements are reduced is now described. Many, if not most, boundary and finite element codes make extensive use of global arrays to store such parameters as nodal coordinates, displacements, tractions, boundary condition codes and associated parameters. With BEAP, extensive use of local arrays is made. The displacements, tractions, etc are stored as local arrays for each element. As each element has exactly 9 function nodes, the storage requirements in memory and on disk can easily be established and the element information is then stored on disk as a direct access file. At any one time it is only required to have two elements (a receiver and an emitter element) in memory, and consequently information for only 18 nodes is required in memory at any one time as compared with 2500 or more if global arrays are used.

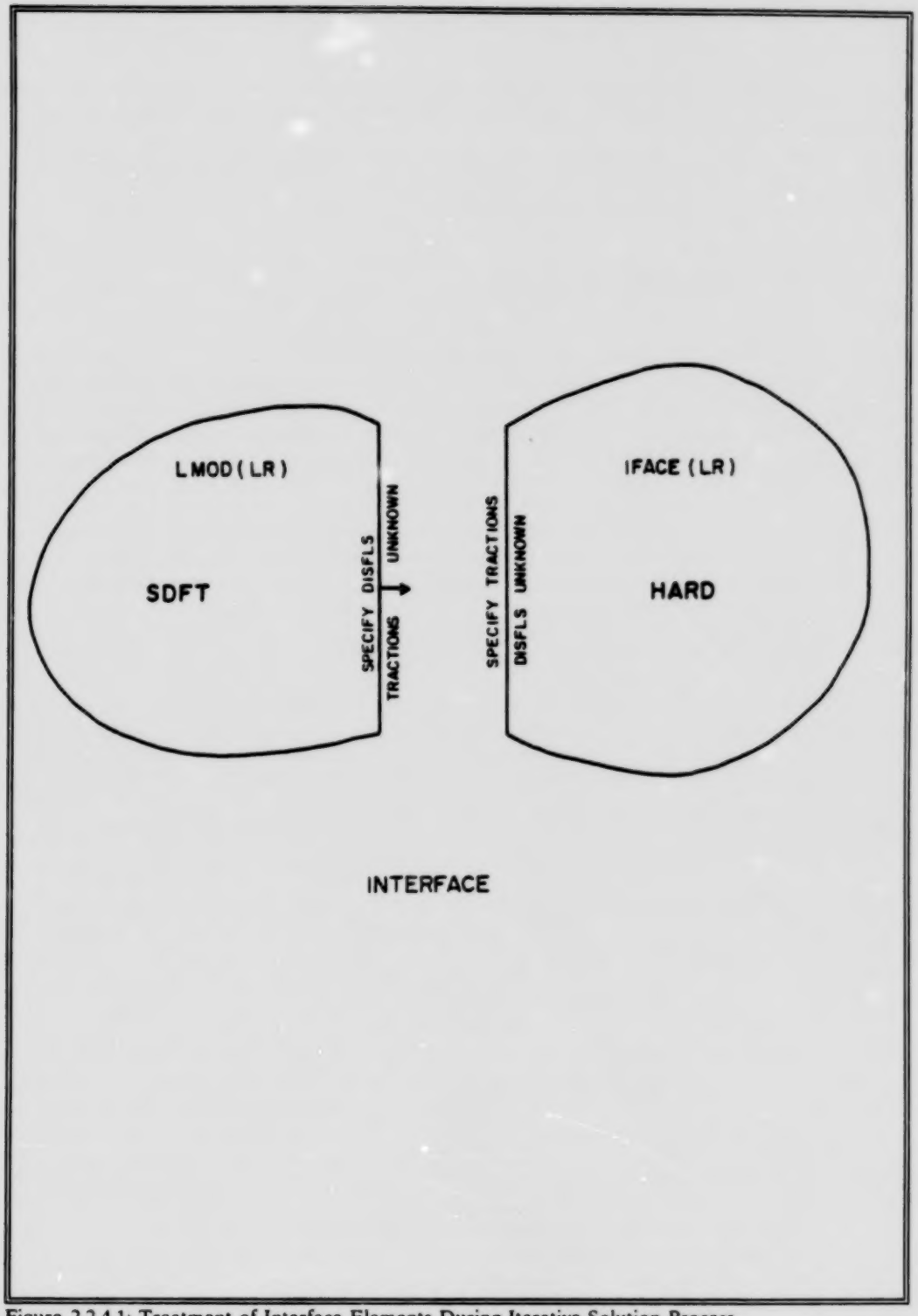


Figure 2.2.4.1: Treatment of Interface Elements During Iterative Solution Process

This whole process would, however, have been impossible or at best very inconvenient if continuous quadratic elements had been used, since nodes are shared with from two to four or more elements.

A disadvantage of the use of local arrays for each element is that disk I/O is increased substantially. If N elements have been defined for a problem, then approximately N**2 disk reads are required for calculation of the coefficient matrix, plus the disk writes required to store the coefficient matrix. However, if M iterations are required to solve the equations, then M*N**2 disk read/writes are required for access of element data alone. Since M has a value of 20 typically, it means that much more disk I/O is required for the solution of the equations than for calculation of the coefficient matrix. As the disk I/O is of a random nature, it results in considerable wear and tear on the disk drive as well as a reduction in solution time. Fortunately, there is an elegant solution to this problem when using an IBM-PC/AT. Use is made of a virtual disk to store the element information. Solution time is reduced by 50% typically, and disk wear and tear is also substantially reduced.

Use of local arrays for each element results in a further benefit for the user. The user can conveniently name each element in the mesh, for example: "Stope 5 hangingwall", or "Stope 4 East side", etc. When results for elements are reported, they are easily grouped into locations which are meaningful to the user.

Each of the basic functions of the boundary element solution process has been built into separate program modules. Examples of these modules are: reading data, checking data, printing data, calculating the coefficient matrix, equation solution and printing element results. On personal computers, these modules may easily be accessed under menu control or by use of batch files resulting in more flexibility to the user.

The biggest single drawback when running BEAP on a personal computer is the speed of the computer itself. Fortunately, though, rapidly changing hardware technology is resulting in ever faster personal computers.

A summary of the program modules making up the BEAP system is as follows

M101 Initialize an entire new database (BEAP and BEAP-DD)
M103 Initialize displacements and tractions

M201 Read in node and element geometry
M202 Check nodes and elements (usually included in M203)
M203 Read in element TYPE definitions and check
M204 Print selected elements or nodes (Detailed print)
M205 Display selected elements to screen
M209 Print all element geometry (batch mode of M204)

M301 Prepare a 3-D view of mesh/elements/nodes

M401 Read in other input data
M402 Print header page and selected information

M501 Apply tractions due to primitive stresses
M502 Calculate lumping maps (BE-BE)
M503 Calculate influence coefficients (BE-BE)
M504 Solve equations (BE)
M507 Calculate benchmarks (BE)
M508 Calculate surface stresses and normal displacements
M509 Convert surface stresses into benchmark values

M601 Print selected element results
M602 Print all element results (batch mode of M601)
M603 Print selected benchmark results

M701 Prepare selected element results for plotting
BEMPOST Post processor for benchmarks

M901 Print nodes and element in data file format
M902 As for M901, with reversed element ordering
M903 Display/modify number of nodes, benchmarks and elements
M905 Load in a new message file

2.3 BEAP-DD

The theoretical background to the displacement discontinuity elements used in this program is well established and well reported, and so is not included here. However, the implementation of this theory into a practical computer program differs somewhat from previous formulations, in that each lump (group of 5 X 5 elements) may have arbitrary orientation. The next sections give a description of the new formulation, followed by various test examples.

2.3.1 Numerical formulation

Using the same summary notation as for the boundary element coefficients, the governing equations for BEAP-DD can be summarized as follows:

$$s = Sd + Kd$$

where
- d is the vector of displacement discontinuities
- s is a vector of stress loads at displacement discontinuity elements
- S is an array of displacement to stress coefficients
- K is an array of displacement to stress coefficients for a regular grid of DD elements

The Kd component represents interaction between displacement discontinuity elements of the same size and orientation which belong to the same (regular) grid. The interactions between displacement discontinuity elements of differing sizes or orientation is included in the Sd term.

2.3.2 Description of lumps

Each lump must be square and contains 25 displacement discontinuity elements. Each lump has, associated with it, two local co-ordinate systems. (figure 2.3.2.1). From this figure, it is seen that an element within a lump may be identified either by its local number (from 1 to 25) or by its local row and column number. For example, element 11 lies in row 4, column 2. The element numbers were chosen so that the first nine elements are in the same relative positions as the function nodes used in BEAP. The remaining element numbers were allocated in a similar fashion. Note, however, that for post processing functions, groupings of the first 4, 8,9 or 13 elements all generate a regular pattern. The intrinsic co-ordinate system (1,2) is used for the shape functions in a similar fashion to those in program BEAP.

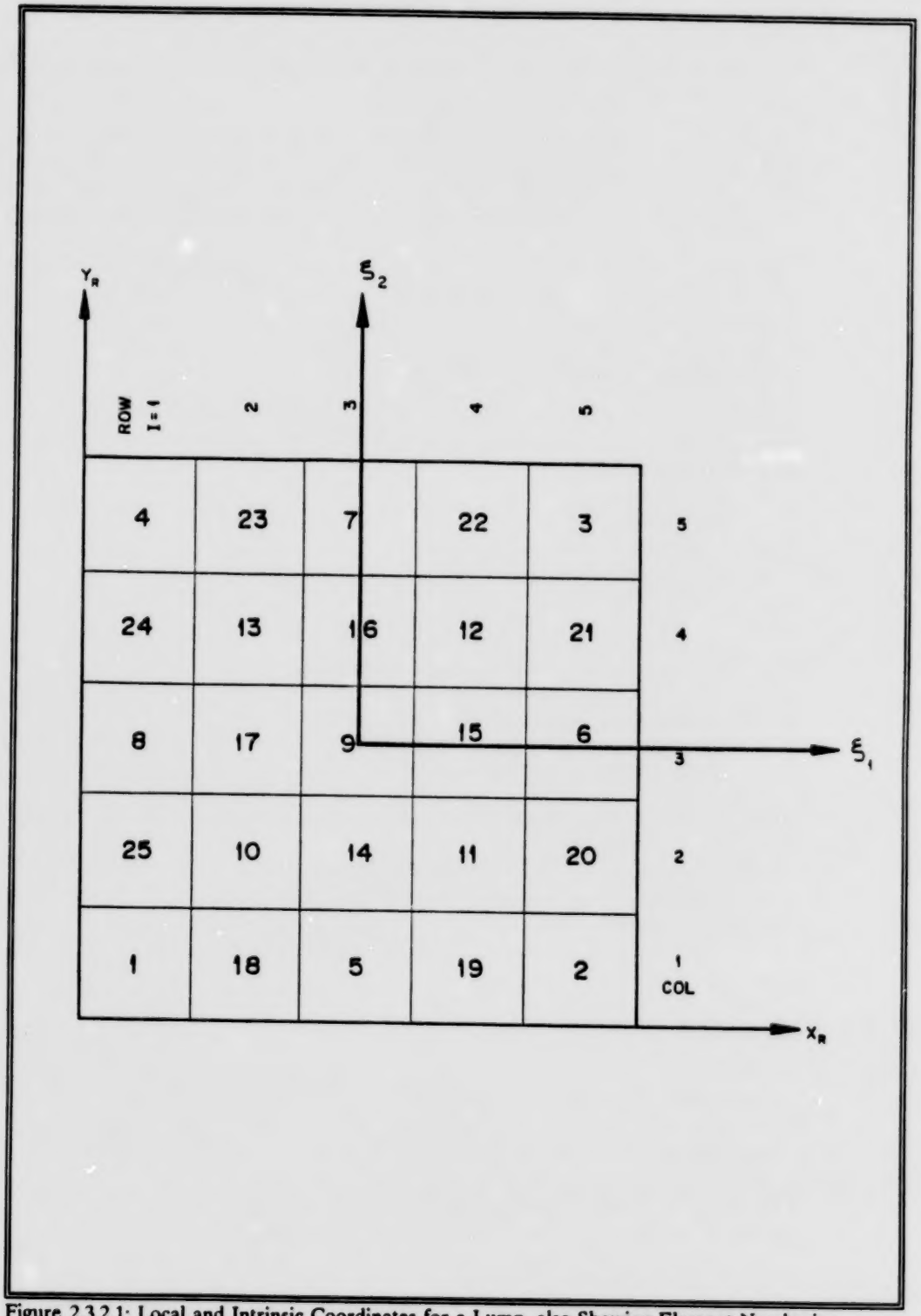


Figure 2.3.2.1: Local and Intrinsic Coordinates for a Lump, also Showing Element Numbering

The local co-ordinate (XR,YR) are used to define the local tractions and displacement discontinuities for each element.

It has been stated that each lump can have arbitrary orientation. Fig 2.3.2.2 shows an examples of two such lumps relative to the global co-ordinate system (XG,YG,ZG). It is also seen from this figure that the origins of the local co-ordinate system (XR,YR,ZR) for a receiver lump and (XE,YE,ZE) for an emitter lump are located at the geometric node 1 used to define the location of the lump.

The global nodes and node numbers defining each lump are defined and read in precisely the same manner as for the non-conforming boundary elements in BEAP, except that only four nodes are needed to define the lump.

2.3.3 Influence coefficients and lumping

In this section a brief description of the governing equations and methods of lumping are presented. A good background can be obtained from Diering (1982), or Crouch and Starfield (1983). Notation used in this section is as follows:

σ	stress tensor
u	displacement vector
d	displacement discontinuity vector
X_R	local co-ordinates for a receiver lump
Y_E	local co-ordinates for an emitter lump
Y	co-ordinate of an arbitrary benchmark point
T	displacement type influence coefficient (Diering 1987)
S	stress type influence coefficients (Diering 1987)
K	displacement discontinuity type coefficient (Diering 1987)
A(R)	transformation matrix from global to local receiver co-ordinates and vice versa
A(E)	transformation matrix from global to local emitter co-ordinates and vice versa

Δ denotes that the influence coefficients have been integrated over an element or lump.

Let the Greek symbols α, β, γ represent displacement, traction or stress components of the receiver lump co-ordinate system. Diering (1982) shows how the equations in this co-ordinate system can be written for dd element i as:

$$\sigma_{\alpha 3}(x_R) = \sum K_{\alpha\beta}(x_R, y_e)\ d_\beta(y_e)$$

for points in the same plane as the grid.

Stresses and displacements at points y outside the plane of the grid are given by:

$$u_\alpha(y) = -\sum d_\beta(y_e) \Delta T_{\alpha\beta}(y, y_e)$$

$$\sigma_{\alpha\beta}(y) = \sum d_\gamma(y_e) \Delta S_{\gamma\alpha\beta}(y, y_e)$$

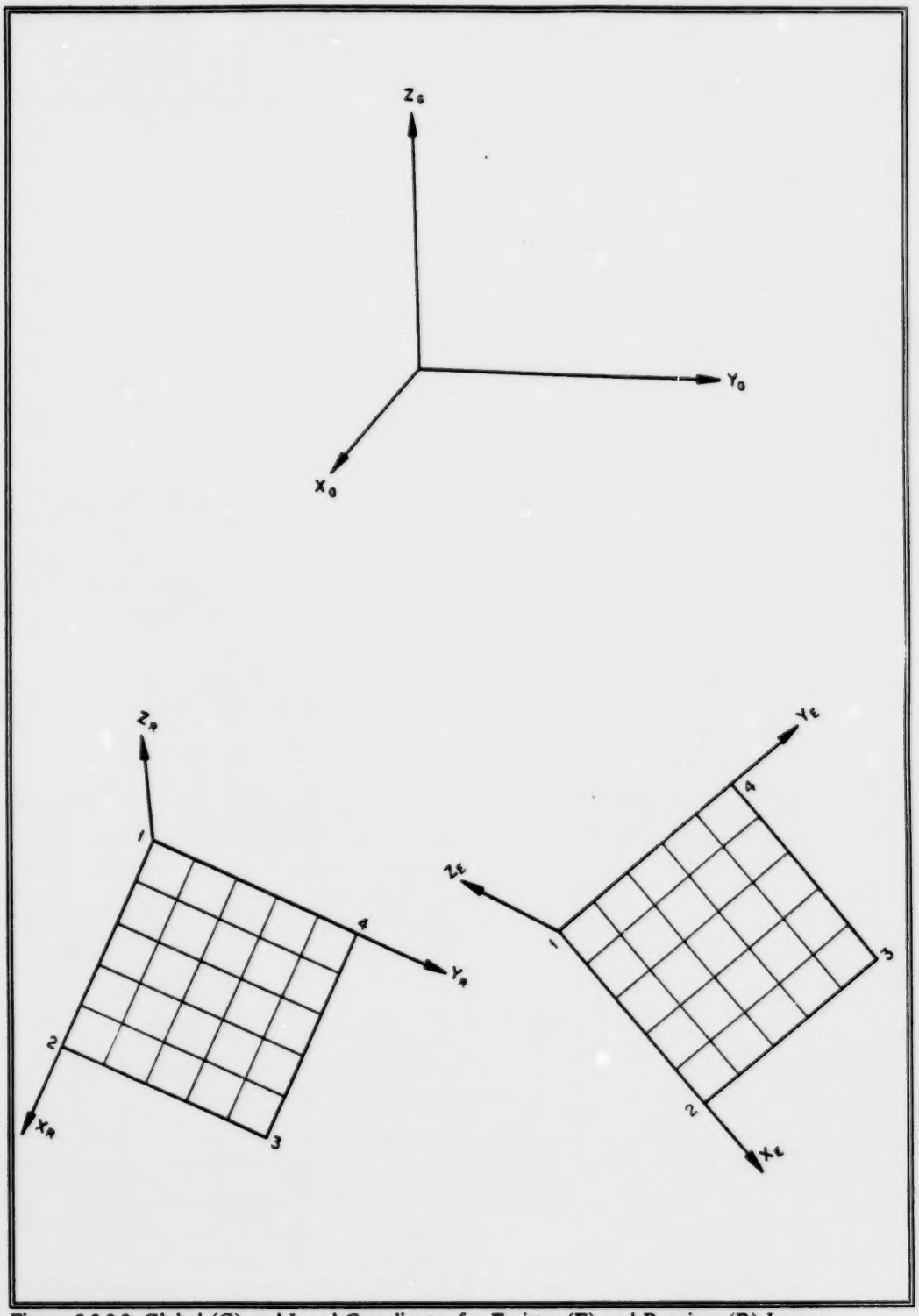


Figure 2.3.2.2: Global (G) and Local Coordinates for Emitter (E) and Receiver (R) Lumps

Now

$$d_\beta(y_E) = A(R)_{\beta i}\ d_i(y_E)$$

where $d_i(y_E)$ are the displacement discontinuity components of the emitter element in global co-ordinates, calculates from the local emitter co-ordinate $d_m(y_E)$ by:

$$d_i(y_E) = A_{\ell i}(E)\, d_\ell .$$

$$\therefore \quad d_\beta(y_E) = A(R)_{\beta i}\left(A_{\ell i}(E)\, d_\ell\right)$$

$$\therefore \quad u_\alpha(y) = -\sum A(R)_{\beta i}\left(A_{\ell i}(E)\, d_\ell\right) \Delta T_{\alpha\beta}(y, y_E)$$

$$\sigma_{\alpha\beta}(y) = \sum A(R)_{\gamma i}\left(A_{\ell i}(E)\, d_\ell \Delta S_{\gamma\alpha\beta}(y, y_E)\right).$$

Consider now the various contributions to a receiver lump and the various elements within that lump. These are shown in Fig 2.3.3.1. In this figure, the variable LINRE is used to distinguish the various types of influence coefficient from one another.

As with BEAP, a map is set up indicating how each lump is related to each other lump. These values are stored in the array LINRE for each lump. The table below gives details concerning different types of coefficient.

LINRE	NCOLR	NCOLE	File	IDD	Name
1	1	1	Not stored	1	Lump-lump DD coefficients
2	1	1	BEMMDD.BE	2	Lump-lump BE coefficients
3	1	25	BEMMDD.BE	2	Lump Elt Be coefficients
4	25	1	BEMMDD.BE	2	Elt lump BE coefficients
5	-	-	-	0	Not used yet
6	-	-	-	0	Not used yet
7	25	25	BEMDD.BE	2	Elt-Elt BE coefficients
8	25	25	Not stored	1	Elt-Elt DD coefficients
9	25	25	Not stored	3	Self coefficients

In this table, NCOLR is the number of receiver "nodes" and NCOLE the number of emitter "nodes". When lumps with the same size and orientation are located in the same plane, it is possible to make use of a standard set of influence coefficients as was the case for program MBEM. Under these conditions, IDD is set to 1. If the emitter and receiver elements both lie in the same lump, then IDD is set equal to 3 and LINRE = 9. However, when two lumps have different size, orientations or are not co-planar, then the influence coefficients between these two lumps have to be evaluated and stored in disk files. Under these conditions, IDD is set to 1 and LINRE has a value of 2,3,4 or 7 depending upon the relative sizes and separations of the lumps (see Fig 2.3.3.1).

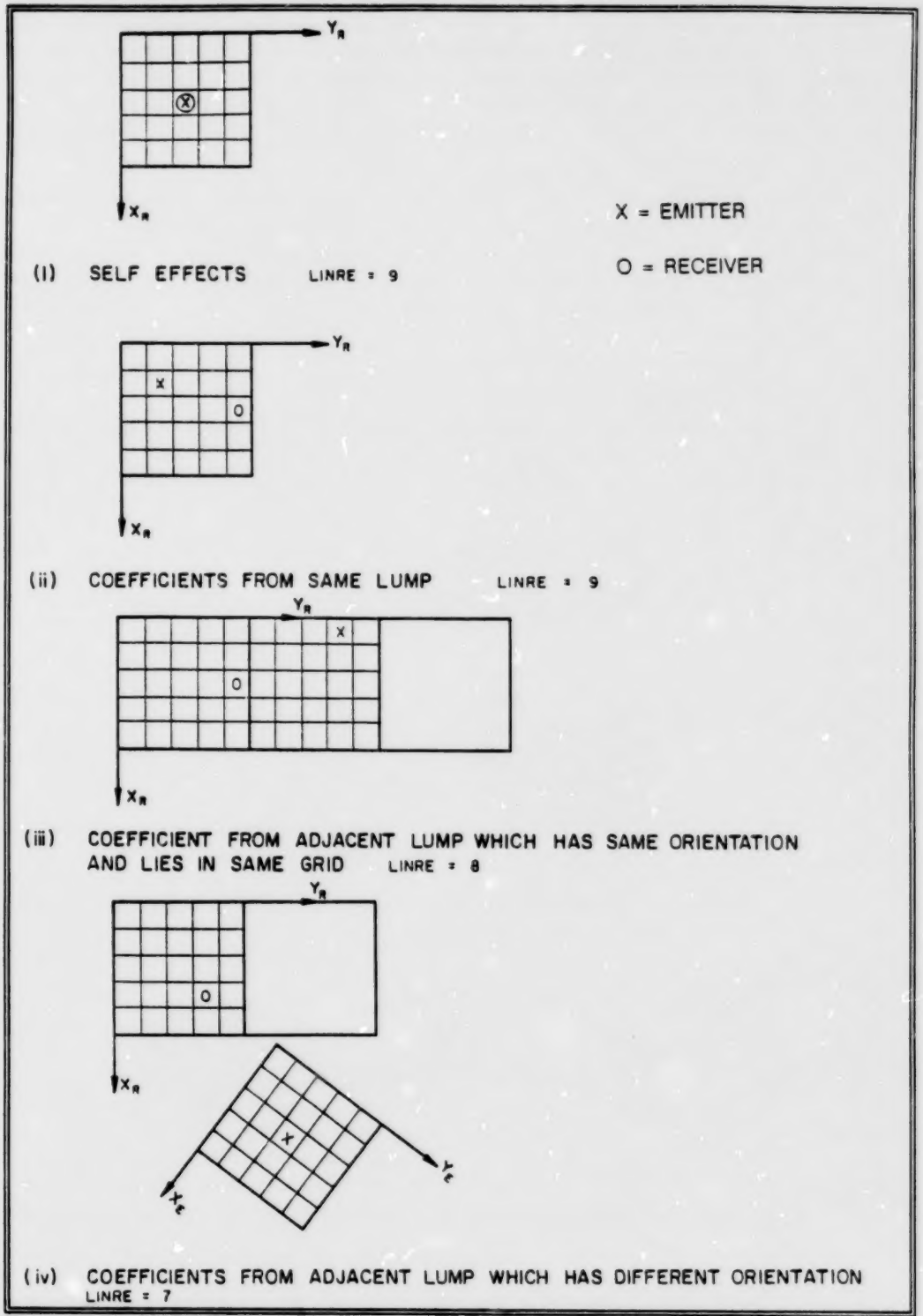

Y_R
X_R
X = EMITTER
O = RECEIVER
(i) SELF EFFECTS LINRE = 9
X
O
(ii) COEFFICIENTS FROM SAME LUMP LINRE = 9
(iii) COEFFICIENT FROM ADJACENT LUMP WHICH HAS SAME ORIENTATION AND LIES IN SAME GRID LINRE = 8
Y_c
X_c
(iv) COEFFICIENTS FROM ADJACENT LUMP WHICH HAS DIFFERENT ORIENTATION LINRE = 7

Figure 2.3.3.1: Different Types of Influence Coefficients Between two Lumps

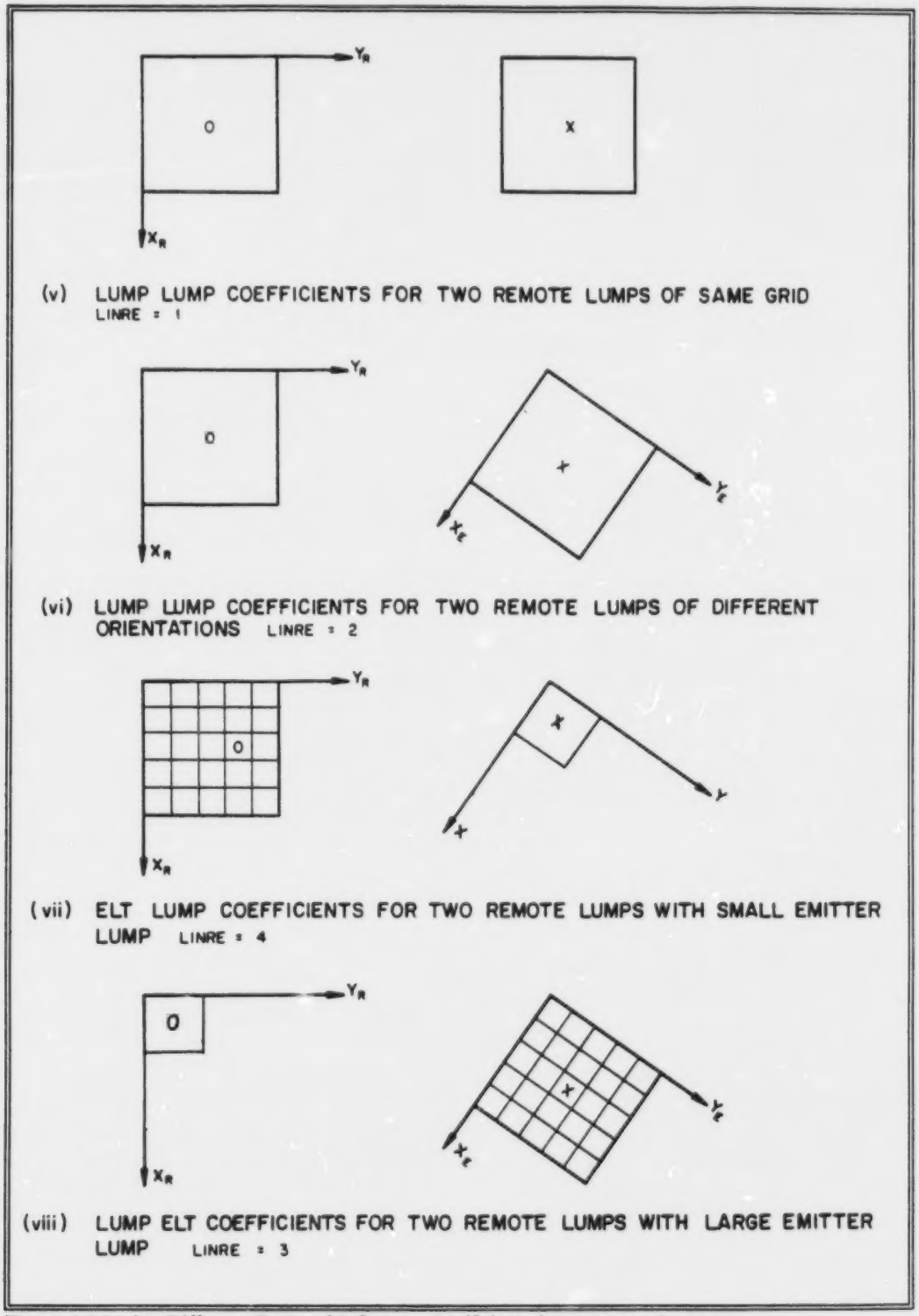


Figure 2.3.3.1 b): Different Types of Influence Coefficients between two Lumps

Obviously, those coefficients which require to be stored on disk will occupy disk space and will also require time to be calculated. Both this space and time are very sensitive to the particular problem being solved. For large problems, the space and time can be prohibitive. The lumping maps and requirements are calculated in program module D501. Output from this module must be studied by the user to see whether sufficient disk space is available to store the coefficients. If not, the problem must be simplified, or the accuracy of the integration reduced.

When IDD = 1, influence coefficients may be calculated in closed form or using numerical integration. When IDD = 2, however, the coefficients must be evaluated using numerical integration. A comparison of numerically integrated and closed form coefficients is shown for the closures for a simple test problem with two lumps (Fig 2.3.3.2). This example clearly demonstrated that the influence coefficients were being correctly calculated for co-planar lumps.

2.3.4 Computational aspects

Figures 2.3.4.1 to 2.3.4.3 below show schematic flow sequences for the input, number crunching and result output phases of BEAP-DD. These figures have been included here to supplement the documentation already provided and also because the same sequencing will be used in the program BEAP-M which is discussed in the next section.

The following program modules make up the BEAP-DD system :-

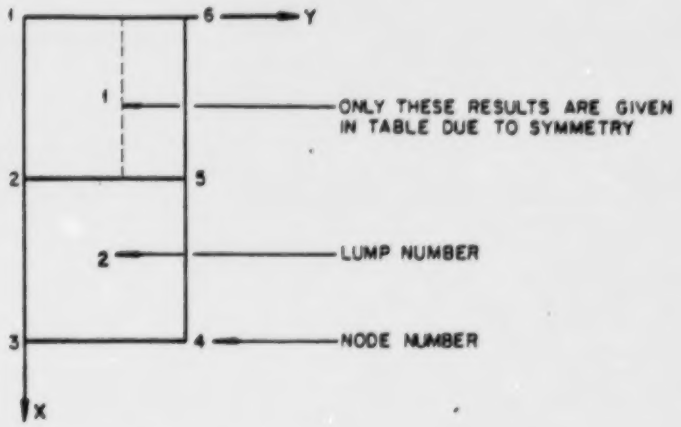
M101 Initialize an entire new database (BEAP and BEAP-DD)
D102 Initialize polygon master file
D103 Initialize closures and stresses

D201 Read in node and DD lump geometry (file D201)
D202 Read in lump grouping (file D202)
D203 Read in lump material properties (file D203)
D204 Print selected DD lumps (summary)
D205 Print selected DD lumps (detailed)
D206 Print out summary of polygons
D207 Read in remaining input (file D207)
D208 Print header information
D209 Print lump summary information (one line per lump)

D301 Generate a grid of lumps

D401 Prepare 3-D view of lump geometry
D402 Prepare polygons for plotting

D501 Calculate lumping maps (DD-DD)
D502 Calculate influence coefficients (DD-DD)
D503 Initialize layout codes prior to first step (D503)
D504 Increment mining step number
D505 Update layout codes for current step (D505)
D506 Calculate tractions due to primitive stresses
D507 Solve equations (DD)
D508 Calculate benchmark stresses and displacements
D509 Calculate energy release rates
D510 Store element results for future reference
D511 Store benchmark values for future reference
D515 Read in a mine layout (in array format) for current step



835	1048	1103
1068	1385	1470
1176	1545	1647
1229	1625	1735
1252	1658	1773

PROGRAM RIDE

835	1048	1102
1068	1385	1470
1175	1544	1646
1229	1624	1735
1252	1658	1772

PROGRAM BEAPDD
CLOSED FORM COEFFIENTS

835	1047	1102
1067	1384	1469
1174	1543	1645
1228	1622	1732
1250	1656	1770

PROGRAM BEADD
INTERGRATED COEFFICIENTS

CLOSURES IN mm

Figure 2.3.3.2: Results for two Lump DD Test Problem

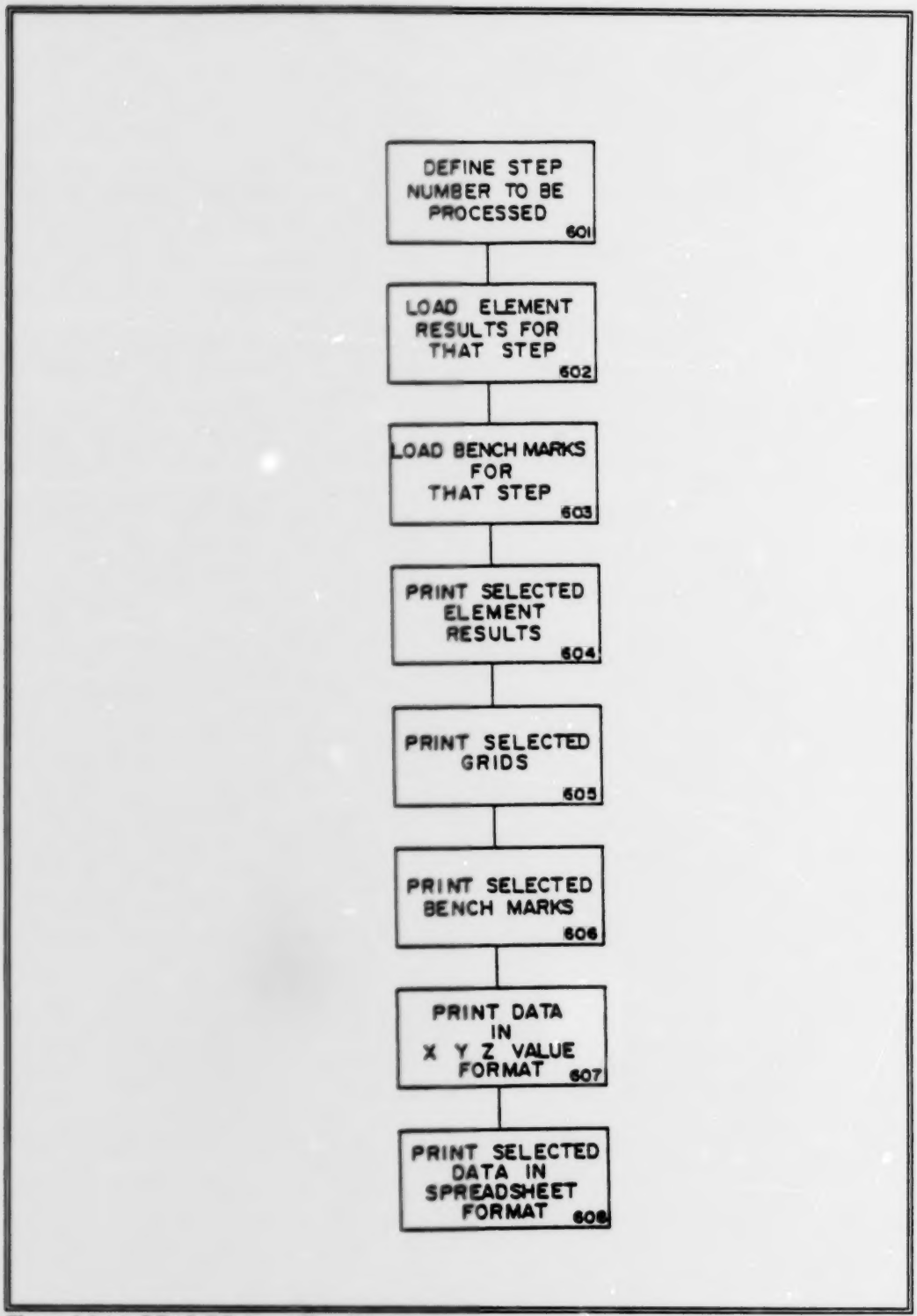


Figure 2.3.4.1: BEAP-DD Input Flow Diagram

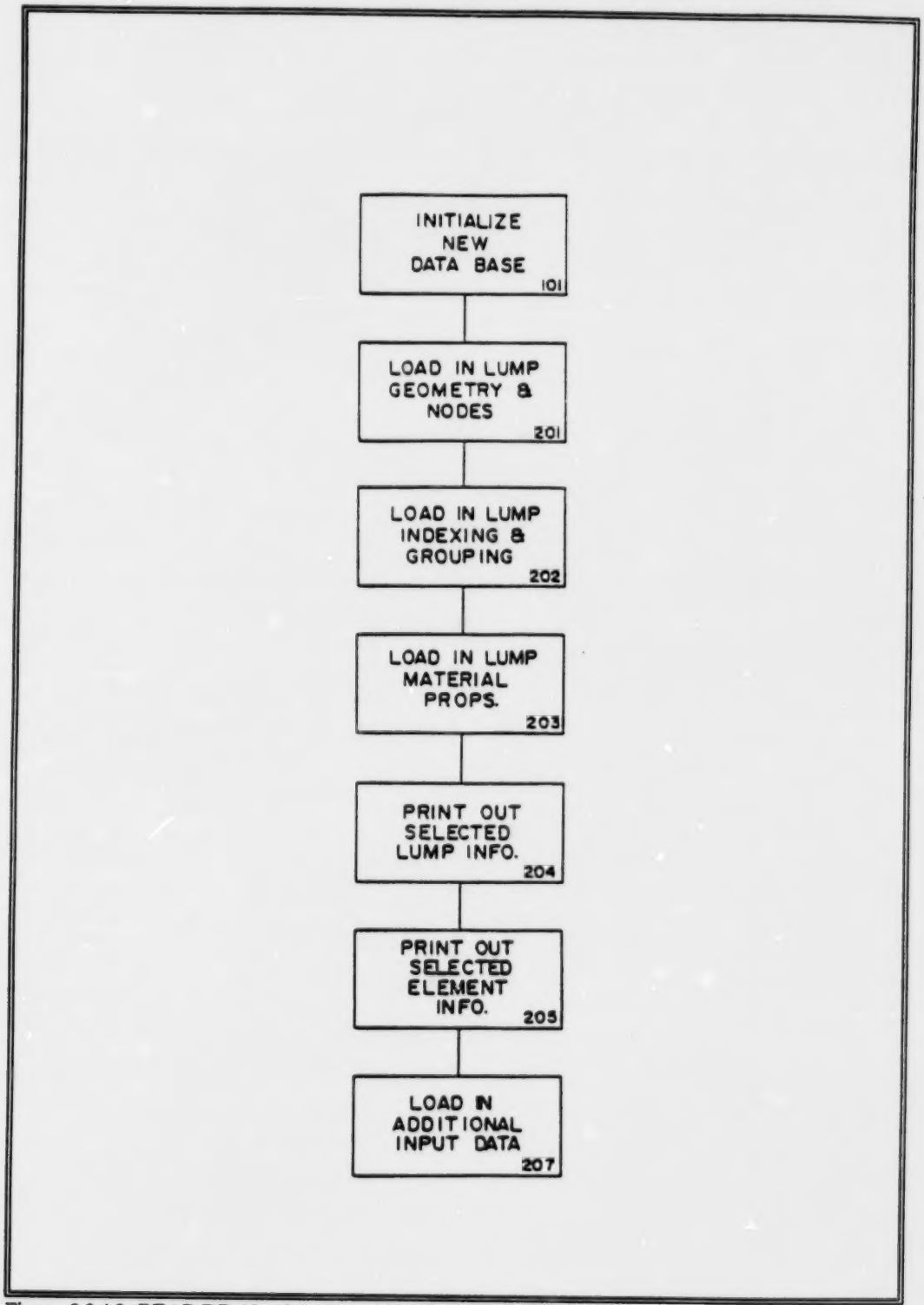


Figure 2.3.4.2: BEAP-DD Number Crunching Flow Diagram

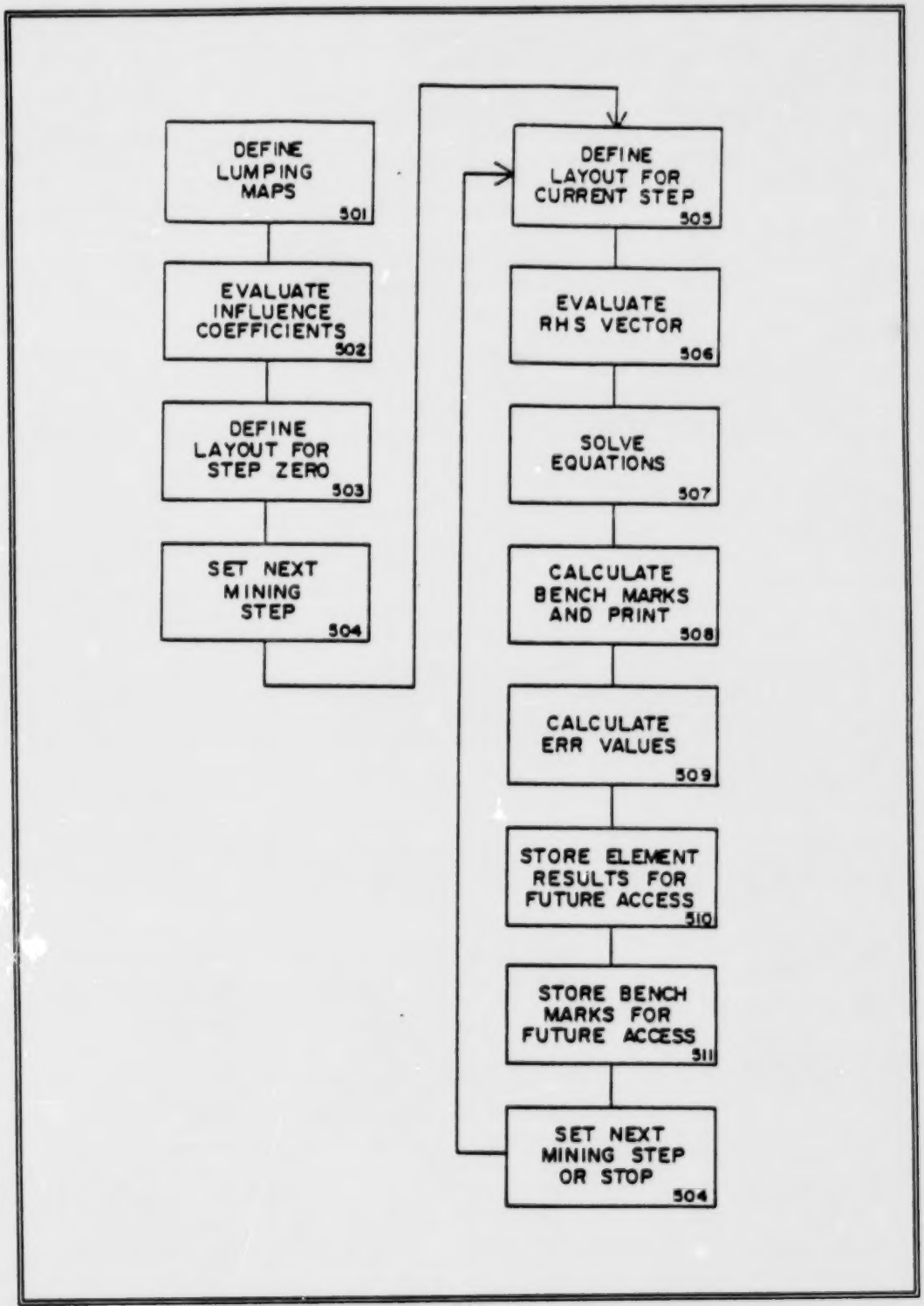


Figure 2.3.4.3: BEAP-DD Post Processing Flow Diagram

D601 Define current step number
D602 Re-load element results
D603 Re-load benchmark results
D604 Print selected element results
D605 Print current layout as a grid
D606 Print selected benchmark results
D607 Write out selected results as PC-MINE extract file
D608 Write out selected results in spreadsheet format

D701 Prepare element data for plotting
D702 Prepare benchmarks for plotting (BEMPOST)
BEMPOST Post processor for benchmarks
D800 Digitizer utilities (separate menu)
D809 Generate a test circular polygon.

D901 Print global nodes and lumps (data file format)
D903 Display/Modify number of nodes/lumps/benchmarks
D904 Set automatic input files on/off
D905 Load in a new message file
D906 Access/display selected messages
D907 Print out selected messages
D908 Set compact of full print mode

The table below gives a summary of the various element types originally planned for BEAP-DD. At this stage, the non-linear solid and partially mined element types have not yet been coded.

TYPE	INTEGER CODE	ASCII CODE	ALTERN-ATIVES	DESCRIPTION
IMIN	1	M	ITOT	Mined out
ISOL	2	+	-	Unmined (Deformable solid)
IBAC	3	B	.	Mined and backfilled
IFLT	4	F	ITEN,ISHR	Fault (pre failure)
INLS	5	N	INLD,INLR	Non-linear solid undergoing down-loading
IDED	6	.	-	Dead element
ITOT	7	C	IMIN	Total closure
ISHR	8	S	ITEN	Shear failed fault
ITEN	9	T	ISHR	Tensile failed fault
INLD	10	D	INLR	Non-linear solid undergoing down-loading
INLR	11	R		Non-linear sold at residual strength
IPAR	12	P		Partially mined element
IZET	13	Z		Zero traction element

Note : Elements currently available in Version 1.0 of BEAP-DD are 1, 2, 3, 4, 6, 7, 8, 9 and 13.

2.4 BEAP-M

2.4.1 Numerical formulation

It has been stated that BEAP-M is a combination of BEAP and BEAP-DD. In achieving the combination of boundary elements and displacement discontinuity elements, the interaction of different boundary and displacement discontinuity element types has to be programmed. In previous sections, details of the boundary elements and displacement discontinuity elements used have been presented. The table below summarizes pertinent information about these element types.

ITEM	Boundary elements	Displacement discontinuity elements
Elements/lump	1	25
Nodes/element	9	4
Shape	Quadrilateral	Square
Degrees freedom per element	27	3
Degrees freedom per lump	27	75
Curved elements	Yes	No

If an example is considered in which boundary elements are used to model some open stopes, and displacement discontinuity elements are used to model surrounding excavations which are assumed to be tabular, then the interaction between all of the above element types would be required to model this situation.

All possible interactions of the above element types are shown in fig 2.4.1.1. The number of possible combinations would have been much bigger had quadrilateral displacement discontinuity elements been included in the formulation. This would have entailed unnecessary additional programming effort, however.

The possible element interactions are also summarized in the next table:

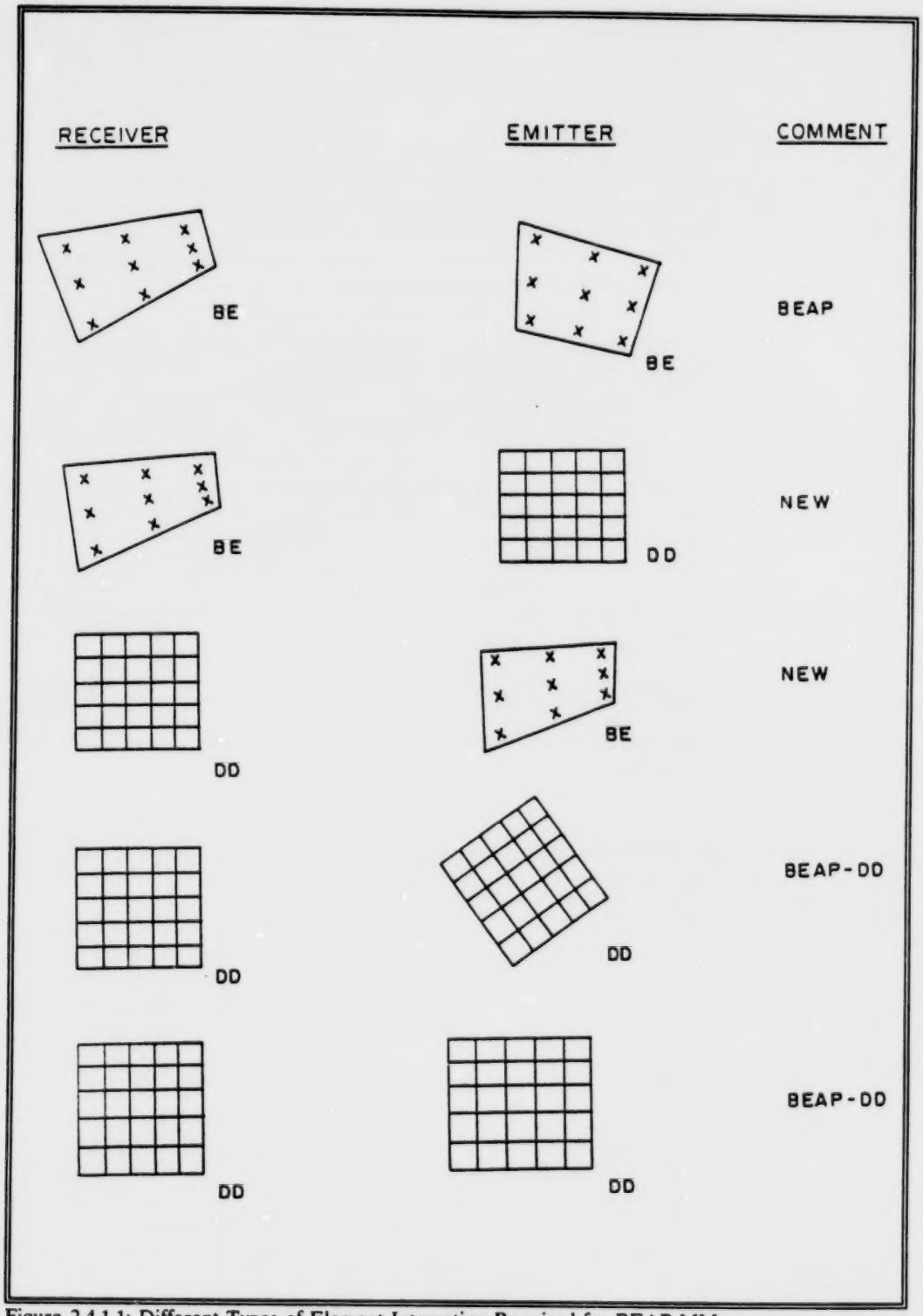


Figure 2.4.1.1: Different Types of Element Interaction Required for BEAP-MM

Receiver Element type	Emitter Element type	Equation format
BE	BE	Tu = Ut
BE	DD	u = Td
DD	BE	s = Su - Dt
DD	DD	s = Sd
DD	DD	s = Kd

where
- u is the vector of BE displacements
- t is the vector of BE tractions
- d is the vector of DD displacement discontinuities
- s is a vector of stress loads at displacement discontinuity elements
- T is an array of displacement to displacement coeffs
- U is an array of traction to displacement coeffs
- S is an array of displacement to stress coeffs
- D is an array of traction to stress coeffs
- K is an array of displacement to stress coeffs for a regular grid of DD elements

Using the above notation, the governing equations for BEAP-M may be simply written as :-

Boundary elements:

$$Tu + Td = Ut$$

Displacement discontinuity elements

$$s = Su - Dt + Sd + Kd$$

2.4.2 Influence coefficients and lumping

The above equations are not strictly correct mathematically, but serve to demonstrate how the various coefficients combine to produce the overall system of equations. By comparison, the equations for BEAP can be written using the same notation as

$$Tu = Ut$$

and the equations for BEAP-DD as

$$s = Sd + Kd$$

All of the above coefficient types have been programmed in either BEAP or BEAP-DD. Most of the programming effort in BEAP-M has been centred around coordinate transformations which have to be introduced into the above equations. The second source of programming effort is that required to

handle the lumping of the above element combinations. When a boundary element is treated as a 'lump', then the displacements and tractions over its 9 nodes are reduced to an average at the centre of the element. Similarly, 25 displacement discontinuity elements will be reduced to a single lump. The number of possible combinations for element-element, element-lump and lump-lump interactions is then much larger than before. The 'lump indexing' for BEAP-M is tabulated below:

Receiver Type	Emitter Type	Coeff Type	Remark	Number of Coeffs.
BE	BE	1-1	lump-lump	1
BE	BE	1-9	lump-elt	9
BE	BE	9-1	elt-lump	9
BE	BE	9-9	elt-elt	81
BE	BE	9-9	touching	81
BE	BE	9-9	self	81
BE	DD	1-1	lump-lump	1
BE	DD	1-25	lump-elt	25
BE	DD	9-1	elt-lump	9
BE	DD	9-25	elt-elt	225
DD	BE	1-1	lump-lump	1
DD	BE	1-9	lump-elt	9
DD	BE	25-1	elt-lump	25
DD	BE	25-9	elt-elt	225
DD	DD	1-1	BE style	1
DD	DD	1-1	DD style	0
DD	DD	1-25	BE style	25
DD	DD	25-1	BE style	25
DD	DD	25-25	DD style	0
DD	DD	25-25	BE style	625
DD	DD	25-25	Self	0

In the above table, the style (DD or BE) refers to the fact that when two displacement discontinuity elements belong to the same grid, then it is not necessary to pre-calculate influence coefficients for that grid. This greatly reduces the number of coefficients requiring calculation, when the displacement discontinuity grids are large (as is standard for programs like MINTAB, MBEM and MINSIM-D).

2.4.3 Equation solution

Equations are solved here in much the same way as described for BEAP and BEAP-DD. However, in this case, the unknown values are both stresses and displacements. Therefore, the rate of convergence of the boundary elements and displacement discontinuity elements is reported separately. It is felt that this will be useful for large problems, particularly if convergence is slow. This is one area in which considerable scope exists for improved iteration schemes. For example, it may be that the displacement discontinuity elements converge twice as fast as the boundary elements for typical problems. If this were the case, then it might prove more efficient to perform two iterations on the boundary elements for every one of the displacement discontinuity element iterations. The program structure would facilitate such further enhancements should they prove necessary in the future.

2.4.4 Benchmarks

Calculation of benchmarks is simply a summation of the effect of the boundary elements and the displacement discontinuity elements, except that the primitive stresses must only be considered once. The way the program has been structured, it would be possible to calculate the benchmark stresses due to either the boundary element component or the displacement discontinuity element component, although this would usually be more than for future research than for day to day type problems.

2.4.5 Computational considerations

Considerable effort has gone into ensuring that the database for BEAP-M is compatible with that for BEAP and BEAP-DD. For example, if one sets up a problem, then the same datafiles can be used to first solve the boundary element part of the problem independently of the displacement discontinuity elements. Results from this run could then be checked and validated. Next, the displacement discontinuity portion of the problem could be run and similarly checked. Given that both 'halves' had been checked, it is then possible to run only those modules which generate the interaction between boundary element and displacement discontinuity elements (ie the DB5.. and BD5.. series of program modules).

A summary of program modules specific to BEAP-M, not included as part of BEAP or BEAP-DD is as follows

DB501 Calculate lumping maps (DD-BE)
DB502 Calculate influence coefficients (DD-BE)
BD501 Calculate lumping maps (BE-DD)
BD502 Calculate influence coefficients (BE-DD)
BD504 Solve equations (MIXED BE+DD)
BD507 Calculate (MIXED) benchmarks

3.0 TEST EXAMPLES

3.1 QUAD

There was no real need for test problems to demonstrate the basic formulation used within QUAD. The theory and application of two-dimensional elements has been very well tested for many years in several disciplines. The real need was for test problems to demonstrate the features of QUAD specific to this project. Two problems are presented below which have been reported in previous annual reports, but which demonstrate some of the non-linear capabilities of QUAD.

3.1.1 Distributed joint elements.

The concept of modelling joints as being uniformly distributed within an element is not new. Numerous cases have been reported in the literature. This report is not concerned with a justification of their use, or with to detailed a theoretical description of their workings. Rather, it was required to demonstrate that the old joint formulation as previously implemented in QUAD had been correctly converted to the new QUAD database structure and that the extension for modelling joint dilatancy was working.

The facility to model distributed joints has been implemented in QUAD for sometime. A description is given by Stacey et al (1982). When the joint spacing within a rock mass is much less than the excavation size, then it is generally impractical to model all joints individually. It then becomes more practical to try to model some of the overall characteristics of actual joint sets.

It is always difficult to obtain input data for any joint modelling exercise. Therefore, the required input data has been deliberately minimized in the interests of simplicity and practicality. The following variables are now required as input for a given joint set:

COHJ	Cohesion of the joint set
PHIJ	Angle of friction of joint set
TENJ	Tensile strength of joint set
ANGLJ	Orientation of joint set
CONJ	Continuity of joint set (1 = fully continuous, 0 = fully discontinuous)
APARM	Dilatancy factor (use 0.001 for no dilation or 1 for a dilation angle equal to the angel of friction PHIJ. Values from 0.001 to 1 give progressively increasing amounts of dilatancy)
RESJ	Residual cohesion of the joint set once it has failed

One example is included here to demonstrate the use of these joint elements. Figure 3.1.1.1 shows the geometry and material properties used in this example. Two elements are used, one of which is a distributed joint element. The second element which is given unit weight is supported by the joint element. If the unit weight of the second element exceeds the joint cohesion, then the joint element fails and failure could continue indefinitely in an unstable manner. If the joints are able to dilate,

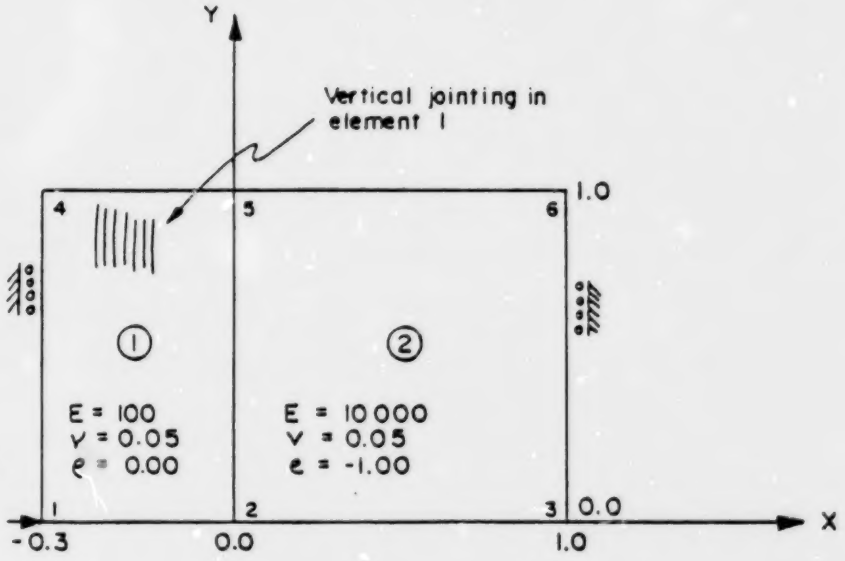


Figure 3.1.1.1: Geometry and Material Properties for Joint Example

however, then as failure progresses, so the joints dilate generating an increasing normal stress across the joints themselves. This increasing normal force then increases the joint shear strength until this is equal to the unit weight of the weight of the second element.

The results from two runs from this test example are tabulated below. The stresses mentioned in this table are those at the centre of element number 1 (ie the joint element). As the joint set modelled in this example is vertical or near vertical, the horizontal stress in the table below is the normal stress acting across the joint set. The shear stresses below are acting across the modelled joints.

Joint Angle	Dilation Factor	Maximum Displ	Horiz. Stress	Shear Stress	No of its Required
95 degrees	.05	.062	-1.35	-1.00	32
90	.05	.407	-1.62	-1.00	>200
90	.20	.119	-1.70	-1.00	81
90	.50	.057	-1.70	-1.00	32
85	.50	.456	-0.07	-1.00	>>200

The following items related to these runs are also of interest

- The shear stress above is correctly maintained at -1.00 being equal to the weight of element 2.
- The run in which a joint angle of 95 degrees was used is kinematically stable and does not need joint dilatancy in order to converge
- By comparison, the run above with a joint angle of 85 degrees is kinematically unstable once tensile failure has occurred across the joint surfaces (the joint surfaces try to move apart from one another). The iterative process fails to converge after 200 iterations and a (relatively) large displacement is achieved.
- For the runs with vertical joints (90 degrees), convergence is achieved when the normal stresses across the joints is -1.70. Convergence is achieved for progressively less iterations as dilatancy increases.

3.1.2 Circular tunnel

The Mohr Coulomb criterion has been use in QUAD for some time. However, previous versions were limited to (initially) no dilatancy and then fixed dilatancy as governed by an associated flow rule. (A good description of flow rules and non-linear finite element modelling is given, for example, by Zienkiewicz (1977)). Neither of these models is well suited to hard rock modelling. A good compromise is achieved by enabling the dilatancy to be fully variable between these two extremes.

Allowance has now been made to accommodate this. However, at this stage, it is up to the user to decide upon the amount of dilatancy which would be required for any given rock mass.

A problem to test the Mohr Coulomb elements was studied with the new QUAD program. The problem is that of a unit circular tunnel subjected to uniform far field stresses of 30 MPa. The material properties are

Cohesion	3.45	Mpa
Phi	30	degrees
Young's modulus	7000	MPa
Poisson's ratio	0.2	

Various runs were carried out to test coarse and fine meshes, quadratic and triangular elements and also the restart capabilities within QUAD. The more interesting runs are summarized in figures 3.1.2.1 to 3.1.2.3 below. The general behavior as modelled by QUAD is correct. Convergence towards the analytical solution as the mesh was made progressively finer was rather slow however, and somewhat disappointing. This slow convergence highlights a property of the linear quadratic formulation for finite elements, namely that these elements are naturally too stiff and generally not well suited to the modelling of extreme cases of failure or plasticity.

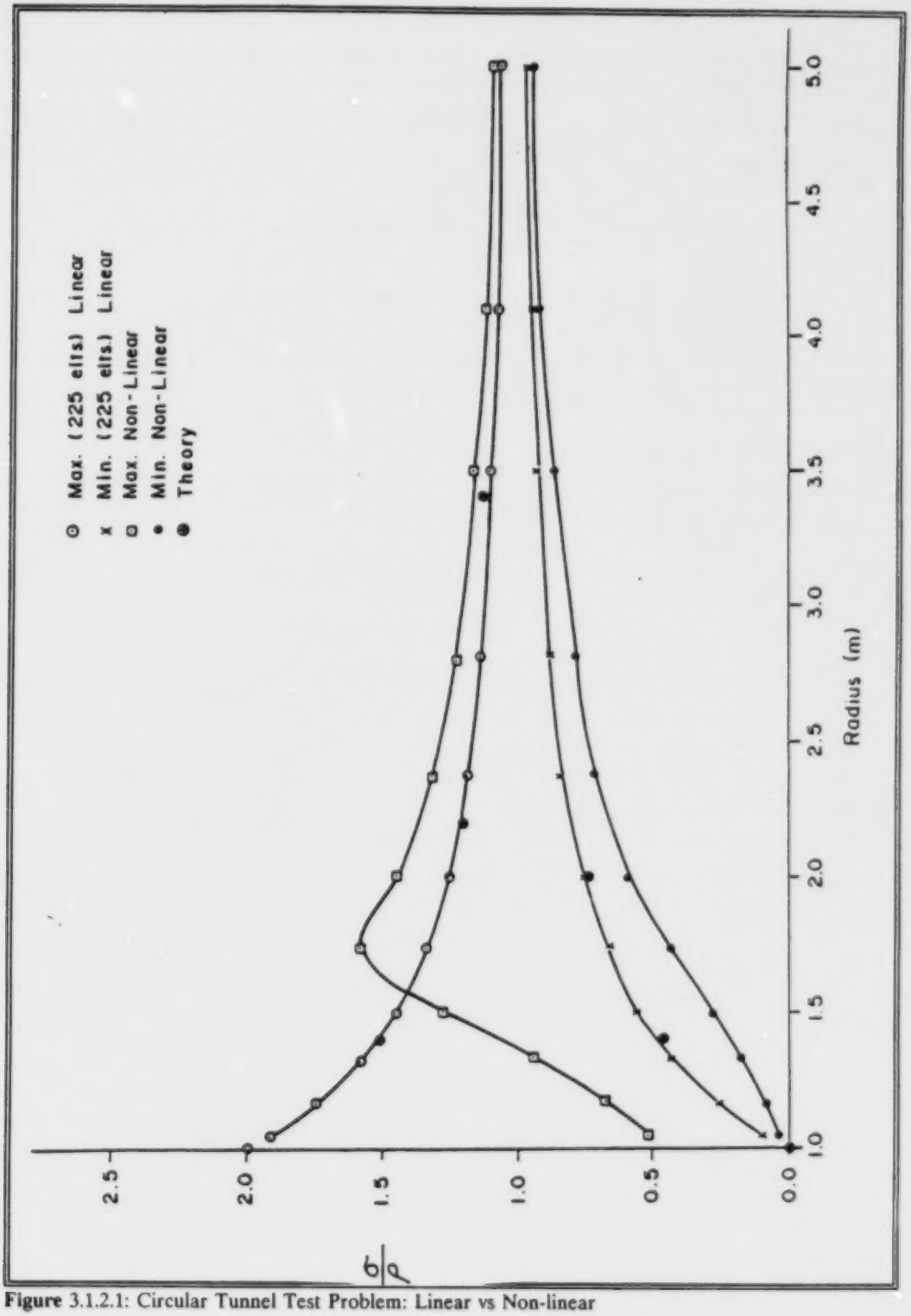


Figure 3.1.2.1: Circular Tunnel Test Problem: Linear vs Non-linear

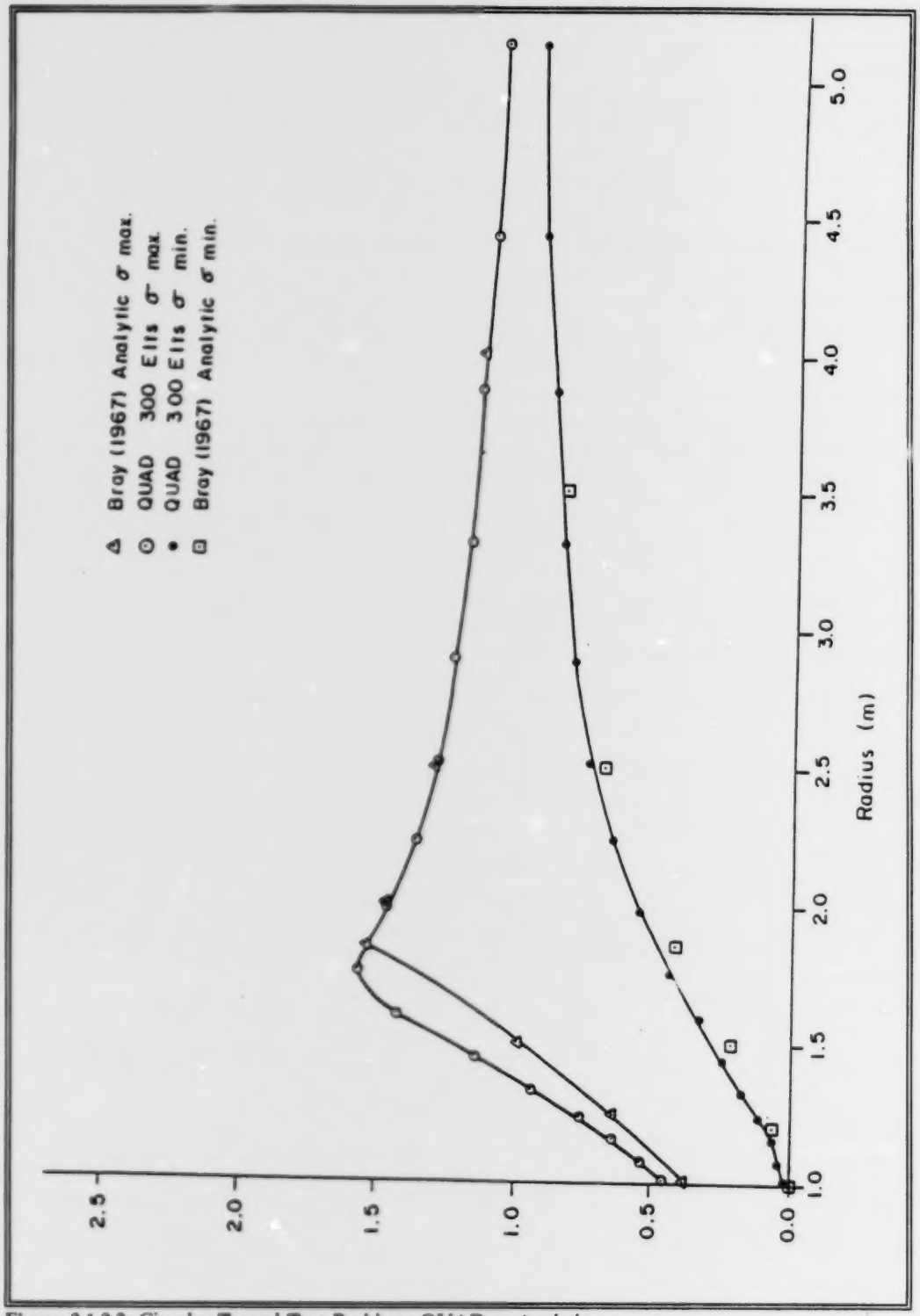


Figure 3.1.2.2: Circular Tunnel Test Problem: QUAD vs Analytic

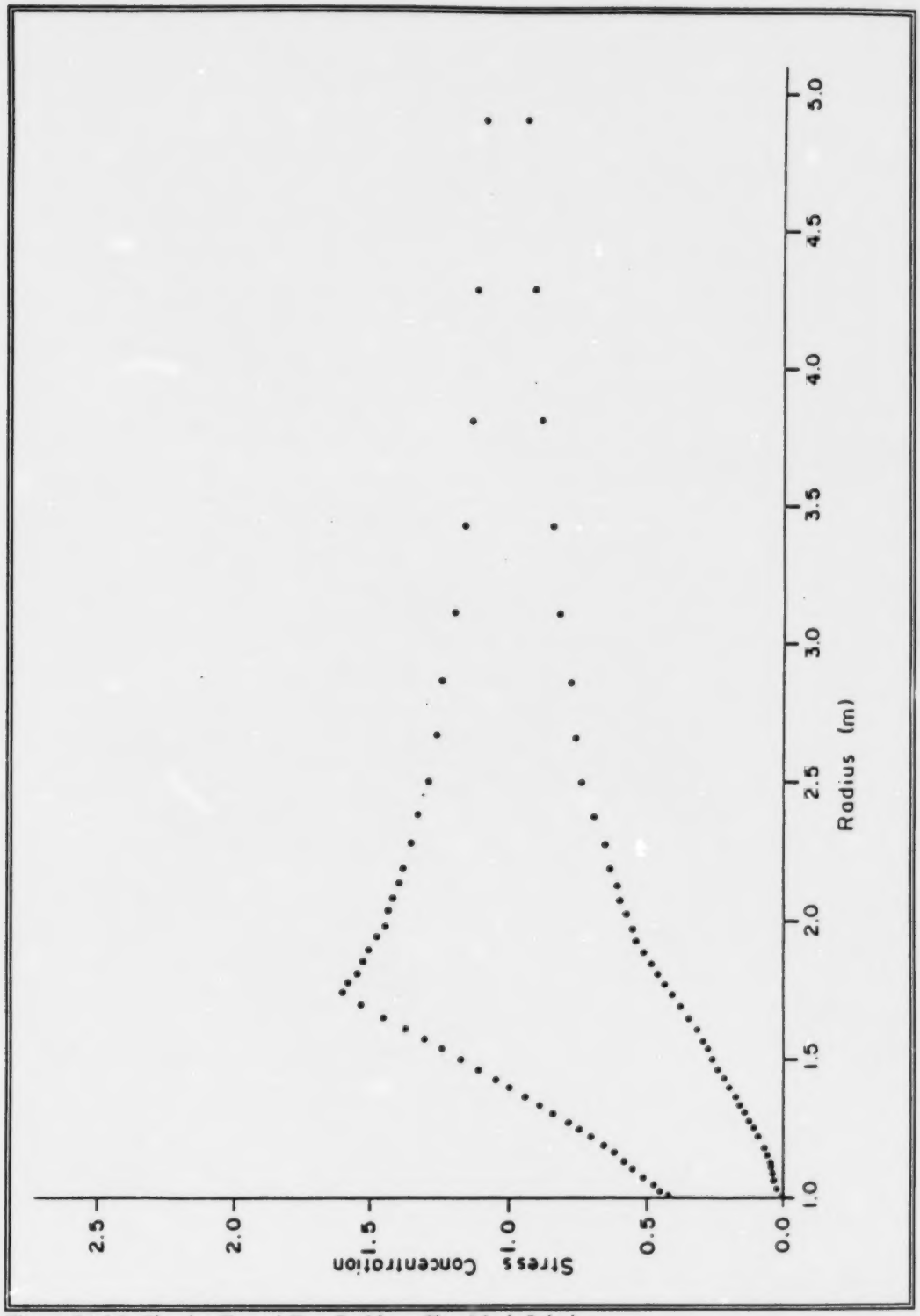


Figure 3.1.2.3: Circular Tunnel Test Problem: Fine Mesh Solution

3.2 BEAP

Numerous examples have been studied as test examples to demonstrate and test non-conforming quadratic elements. Three examples are shown if figures 3.2.1 to 3.2.3. Other examples are described in more detail below.

3.2.1 Two-bonded cubes

This problem was chosen for several reasons. Firstly, it was to demonstrate that the program BEAP could model non-homogeneous problems consisting of more than one subregion. Secondly, it was to show that non-homogeneous problems could still be solved efficiently with an iterative equation solver and, thirdly, the bonded cube problem was chosen to see how well the rapid variation of displacements across the interface would be modelled.

The geometry for the problem is shown in Figure 3.2.1.1. The lower and upper cubes had Young's Moduli of 100 MPa and 10 MPa and Poisson's ratios of 0.2 and 0.3 respectively. A uniform traction of -10 MPa was applied to the top of the upper cube. Normal displacements were prevented in elements 1,3,4,5,8,9 and 10 so as to simulate plane strain conditions again for comparison with two-dimensional solutions. Unfortunately, a two-dimensional boundary element formulation capable of modelling multiple subregion problems was not readily available, so comparison was made with the QUAD program The problem was modelled with 200 square elements and 231 nodes.

A comparison of results (Fig 3.2.1.2) is excellent everywhere except in the close proximity of the interface. It is felt that the difference between the two solutions is due partly to errors in both the finite element and the boundary element solutions, with the true solution being somewhere between the two.

3.2.2 Tunnel with square cross section

The tunnel was chosen to be long, so that comparison could be made with two-dimensional formulations and for comparison of stresses and displacements at interior points. The geometry for the two-dimensional problem is a simple square cross section. The three-dimensional problem has a length: width and length:height ratio both equal to 10:1 so that the conditions close to the centre of the tunnel are very close to plain strain conditions. Details for material properties, loading and geometry are as follows:

Height	10	m
Width	10	m
Young's Modulus	50 000	MPa
Poisson's Ratio	0.25	
Vertical stress at infinity	100	MPa
Horizontal stress at infinity	50	MPa

In particular, Figure 3.2.2.1 shows a comparison of horizontal displacements for this problem using different types. It is clear from this figure that the non-conforming elements perform far better than the quadratic elements over most of the range ABC. It is only close to B (the corner or edge of the tunnel) that the fit is not good. It is also felt that this example demonstrates that 1 non-conforming element per excavation face will generally be adequate for fairly accurate modelling.

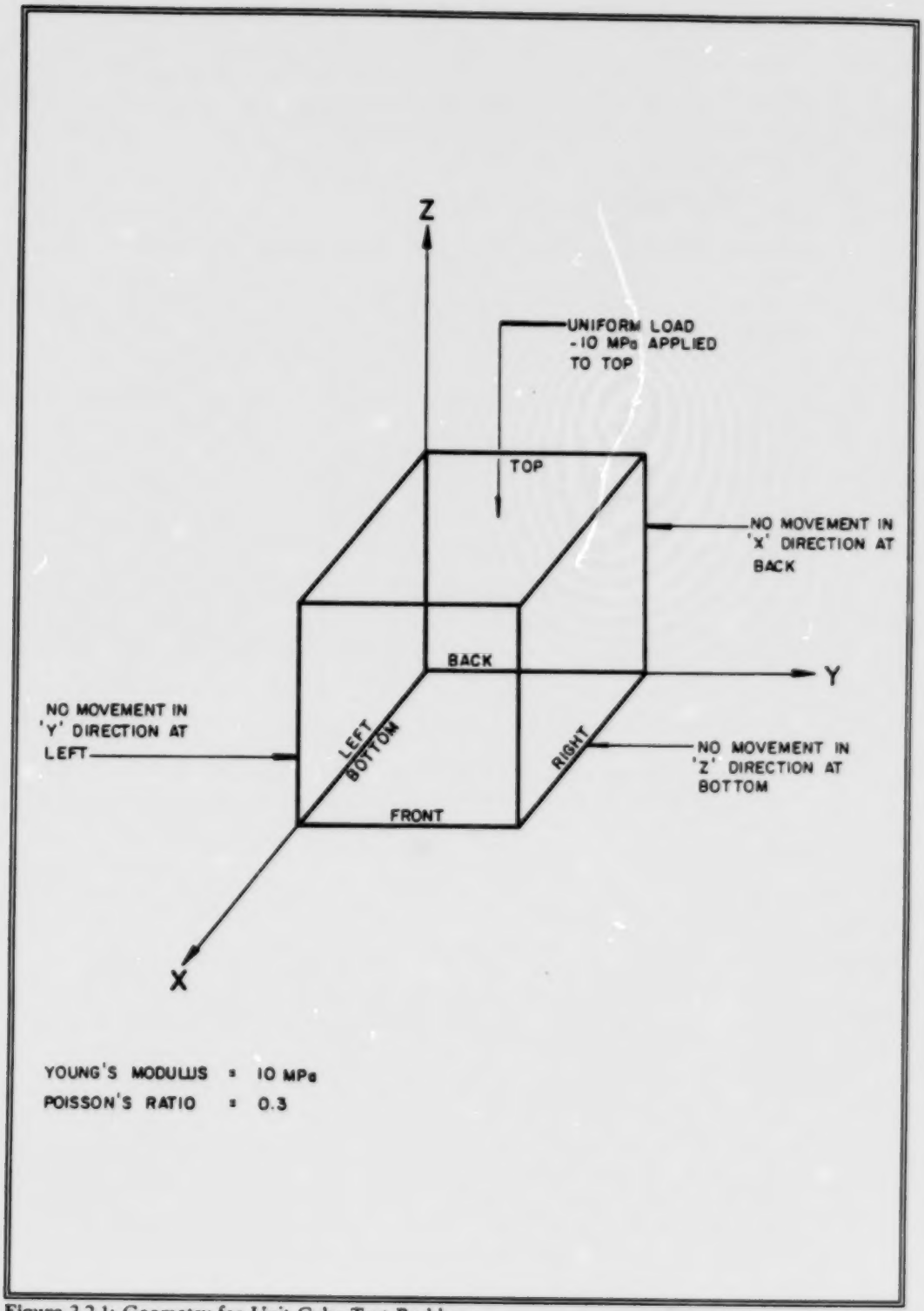


Figure 3.2.1: Geometry for Unit Cube Test Problem

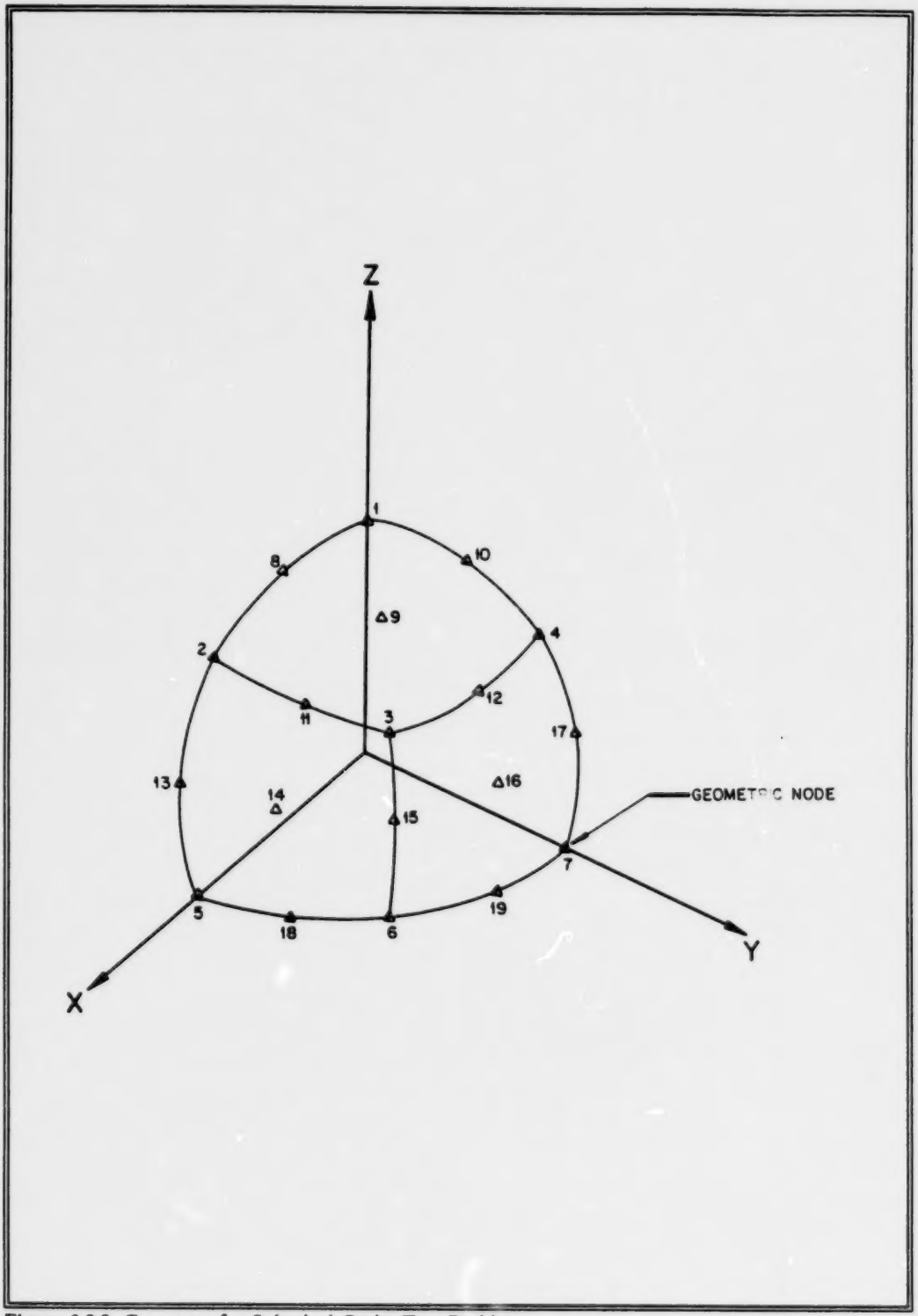


Figure 3.2.2: Geometry for Spherical Cavity Test Problem

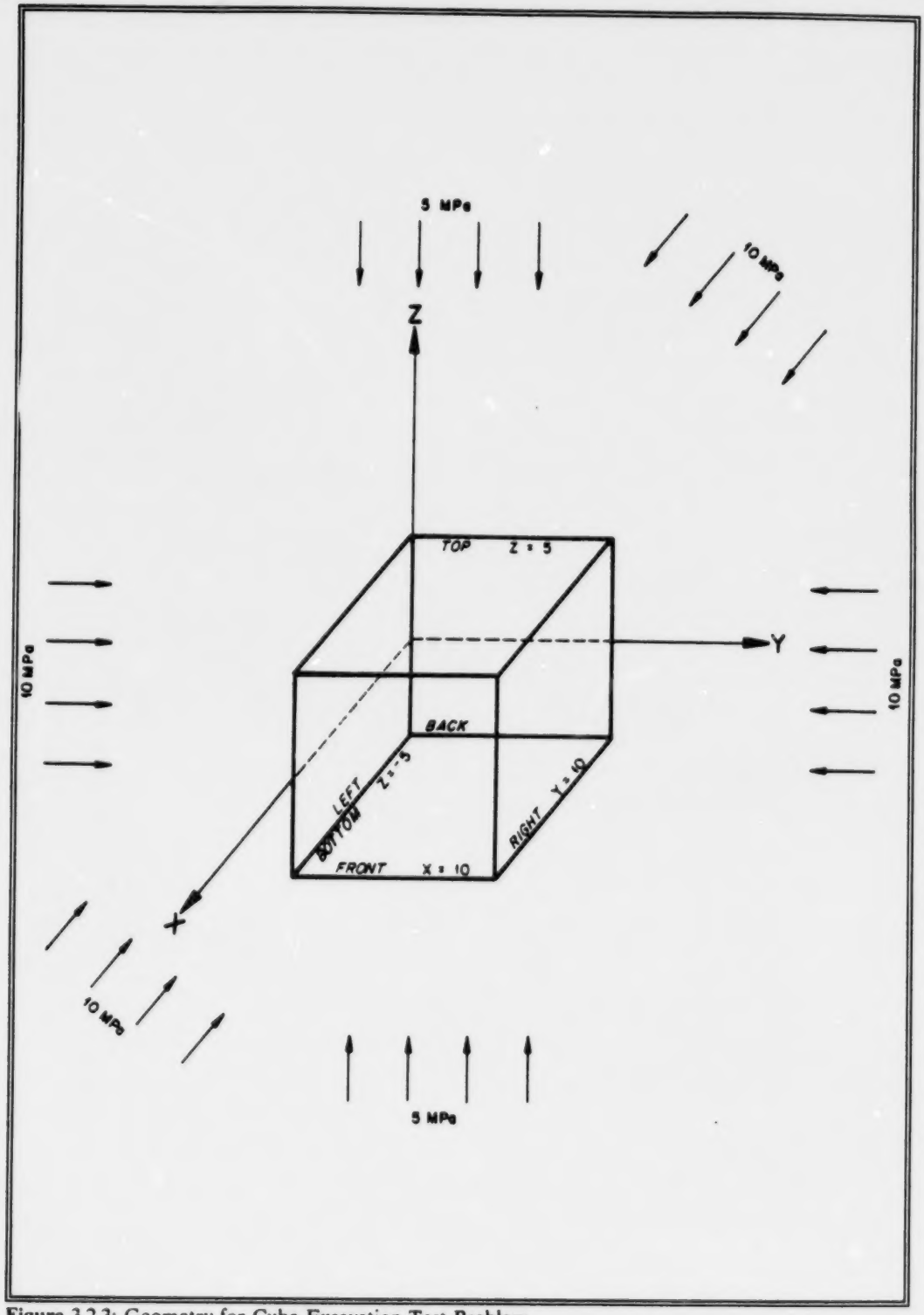


Figure 3.2.3: Geometry for Cube Excavation Test Problem

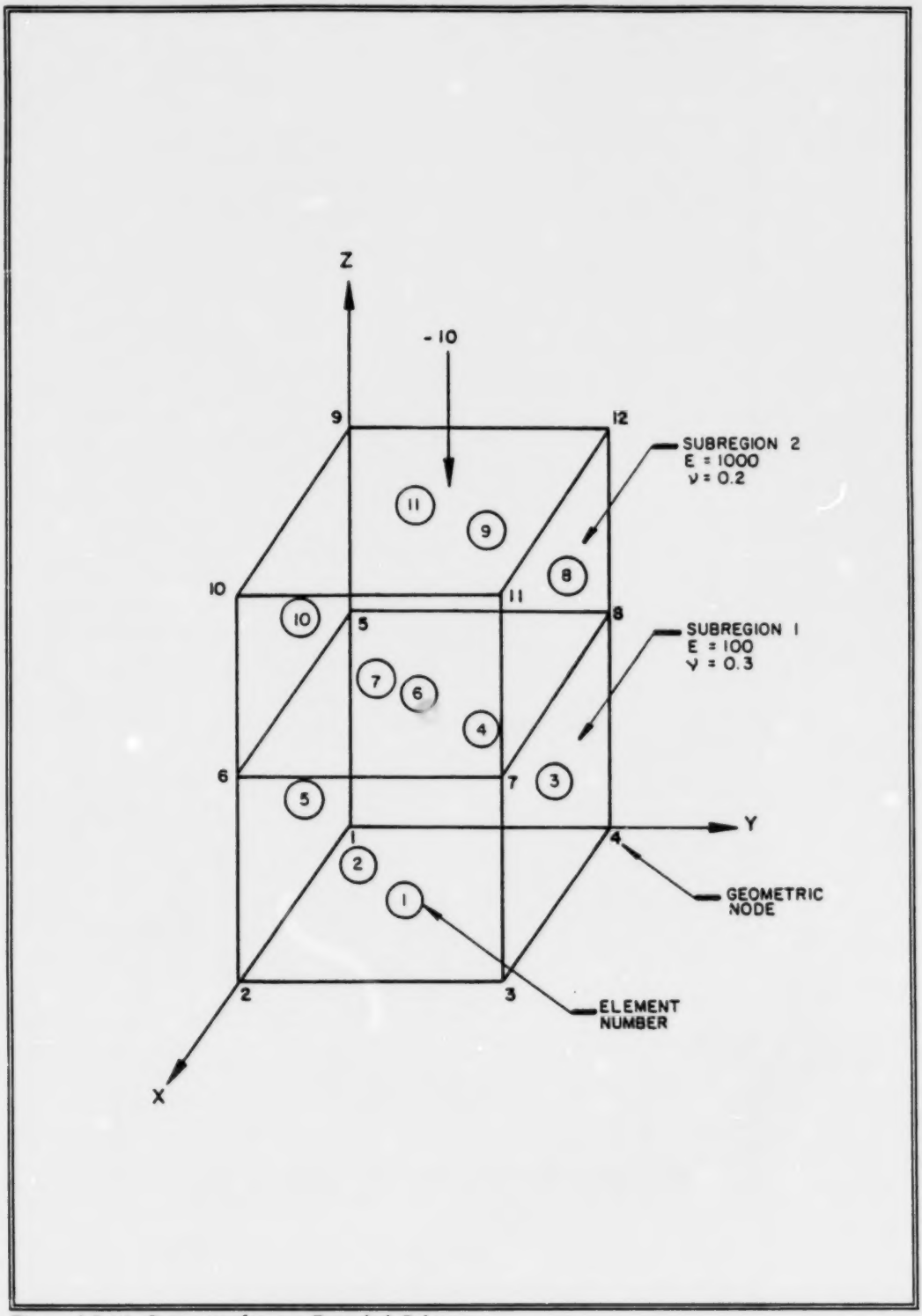


Figure 3.2.1.1: Geometry for two Bounded Cubes

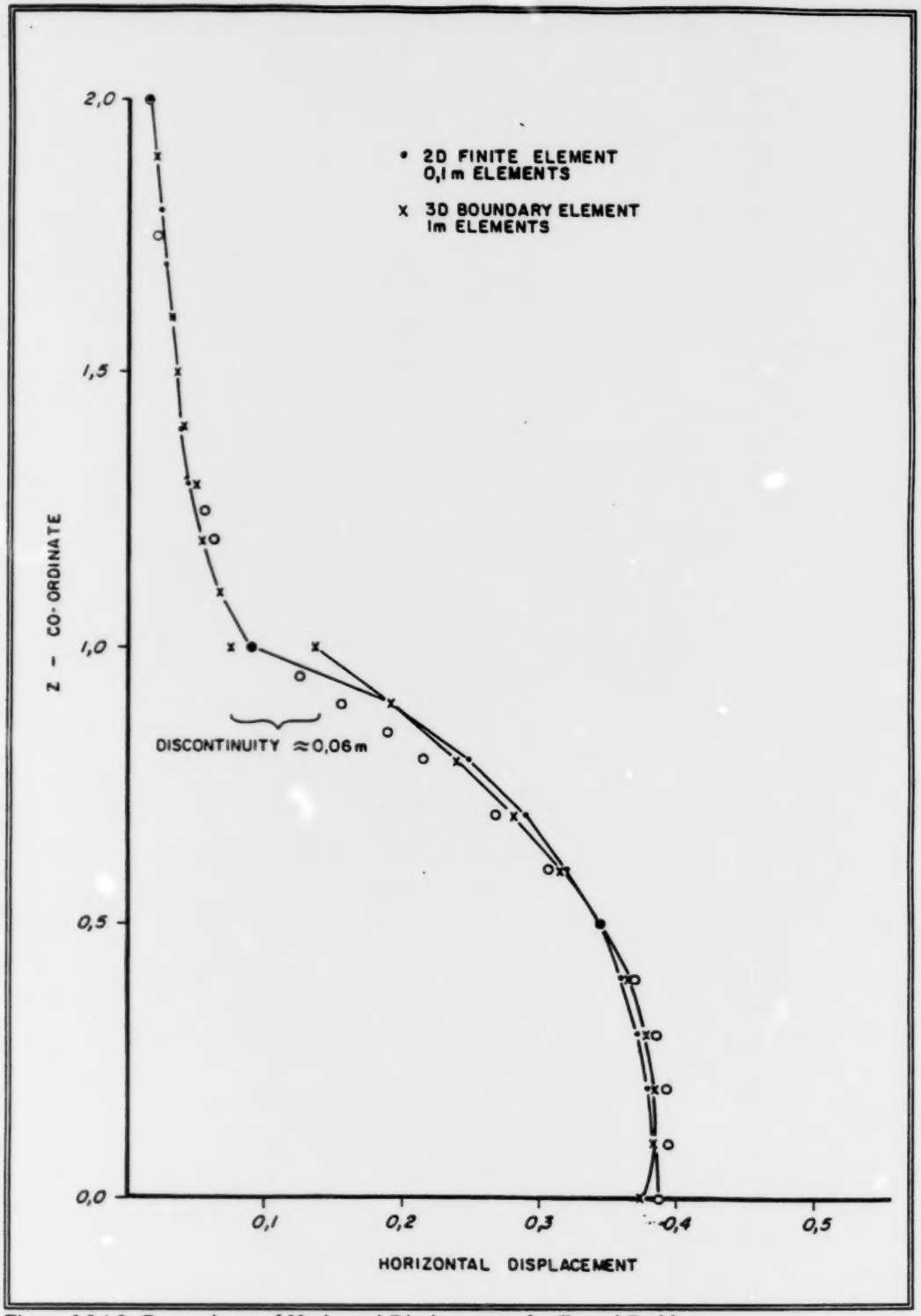


Figure 3.2.1.2: Comparisons of Horizontal Displacements for Tunnel Problem

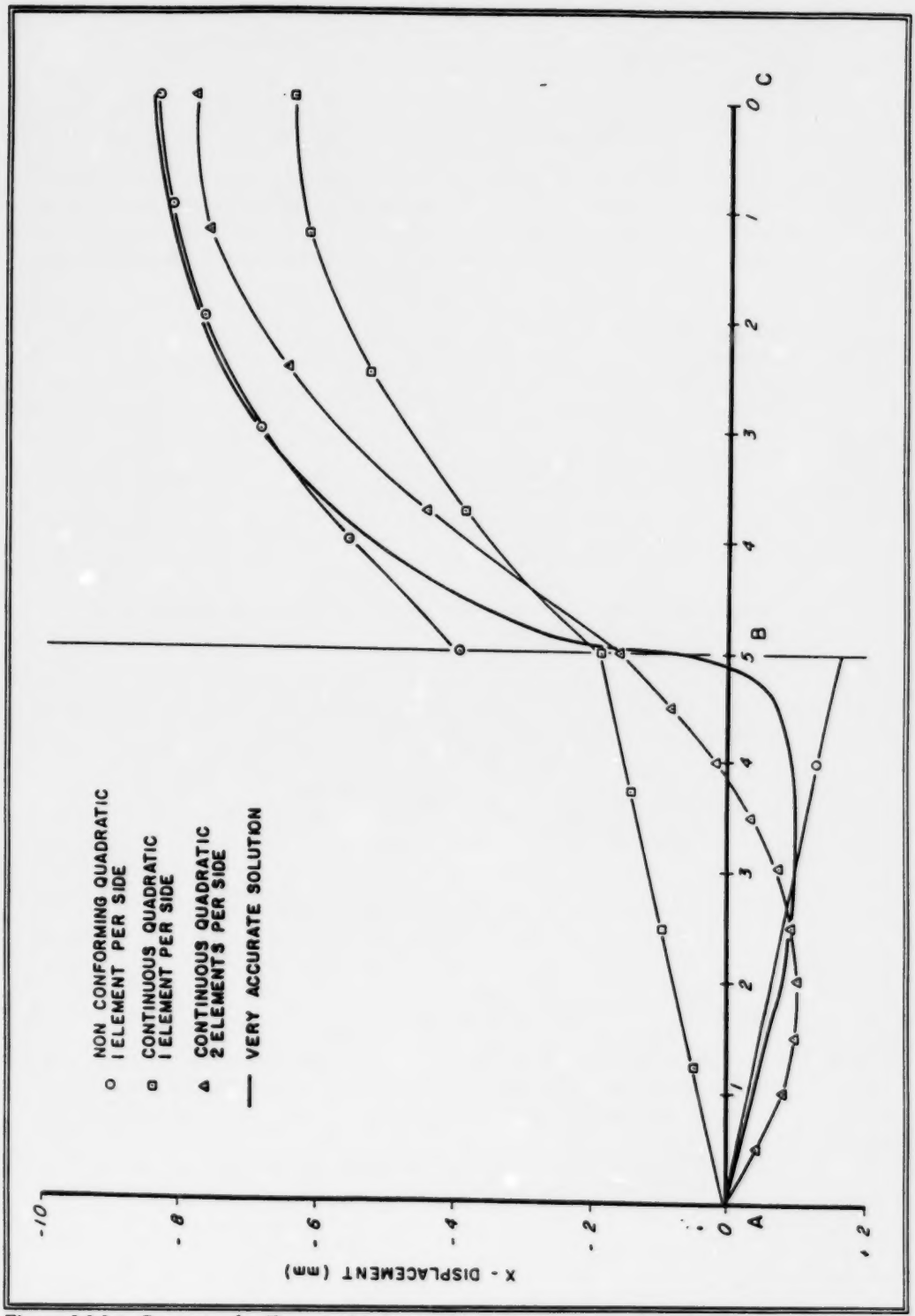


Figure 3.2.2.1: Geometry for Tunnel with Square Cross Section

3.3 BEAP-DD

3.3.1 Small folded seam

This series of runs was chosen, so that comparisons could be made between two- and three-dimensional solutions. The lumps and geometry are shown in Figure 3.3.1.1. It can be seen from this figure that the geometry will be such that approximately two dimensional plane strain conditions are achieved at the centre of lumps 5 and 6. Materials properties and loading for these runs were:

Young's Modulus E	100	MPa
Poisson's ratio	0.3	
Block size	4	m
Vertical applied stress	5	MPa
Horizontal applied stress	10	MPa
Total number of lumps	10	
Total number of elements	250	
Angle of intersection	135 for run 1 90 for run 2 45 for run 3 60 for run 4	

Parameters relating to the numerical integration of influence coefficients for typical elements at the intersection are summarized below:

Run	Angle of Intersection	Highest Separation Ratio	Order of Integration
1	135	0.85355	4
1	90	0.49999	5
3	45	0.146447	20
4	60	0.249998	10

This information is shown schematically in Fig 3.3.1.2.

Comparison of results for 2D and 3D runs

It was hoped and expected that the closures and rides along the centre cross-section through lumps 5 and 6 would be reasonably close to the two dimensional solutions. In general the 3D results should be slightly less than the corresponding 2D results.

Comparisons of the results for reefs intersecting at 135, 90, 60, and 45 are shown in Figs 3.3.1.3 to 3.3.1.6. The agreement in every case is good.

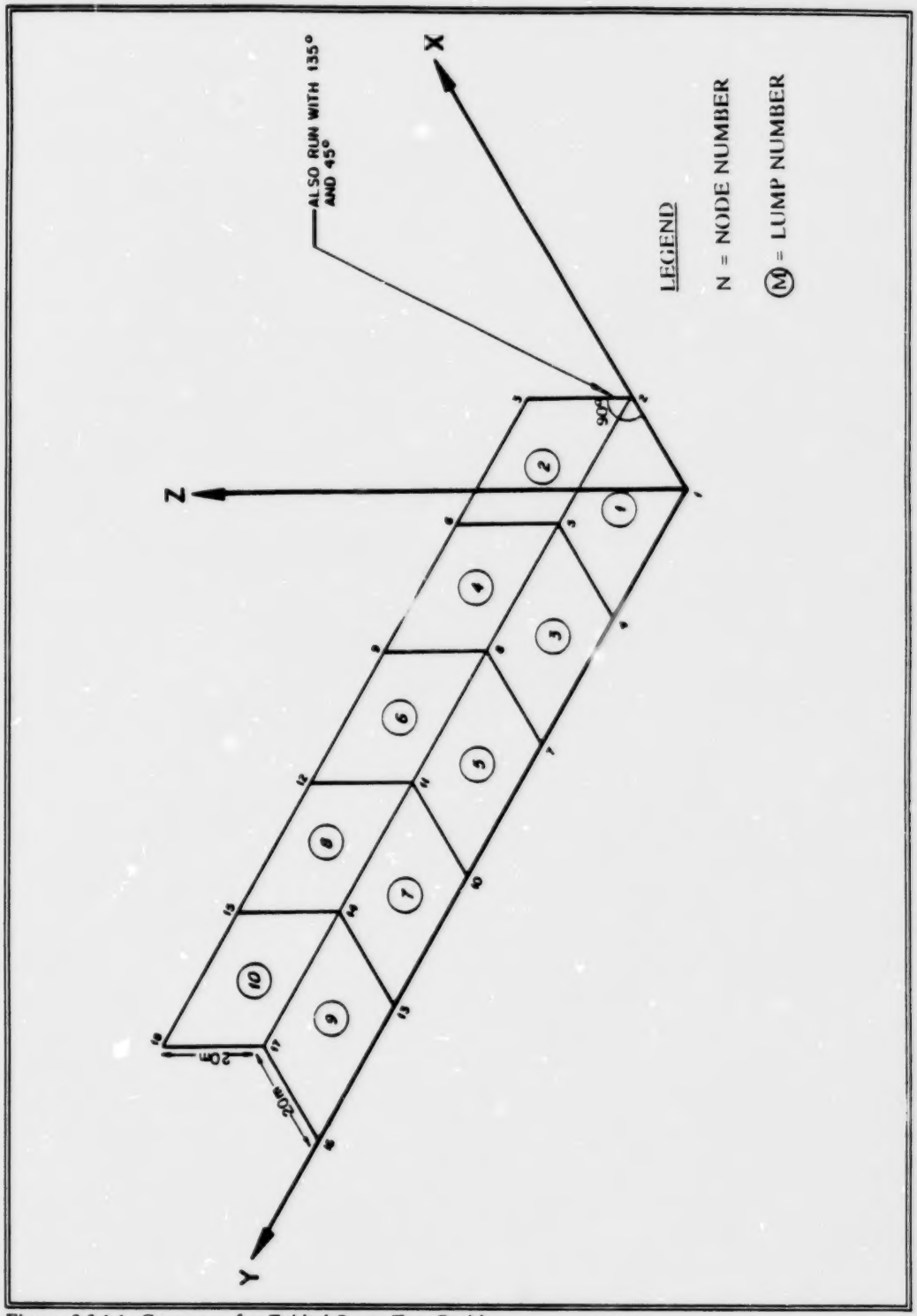


Figure 3.3.1.1: Geometry for Folded Seam Test Problem

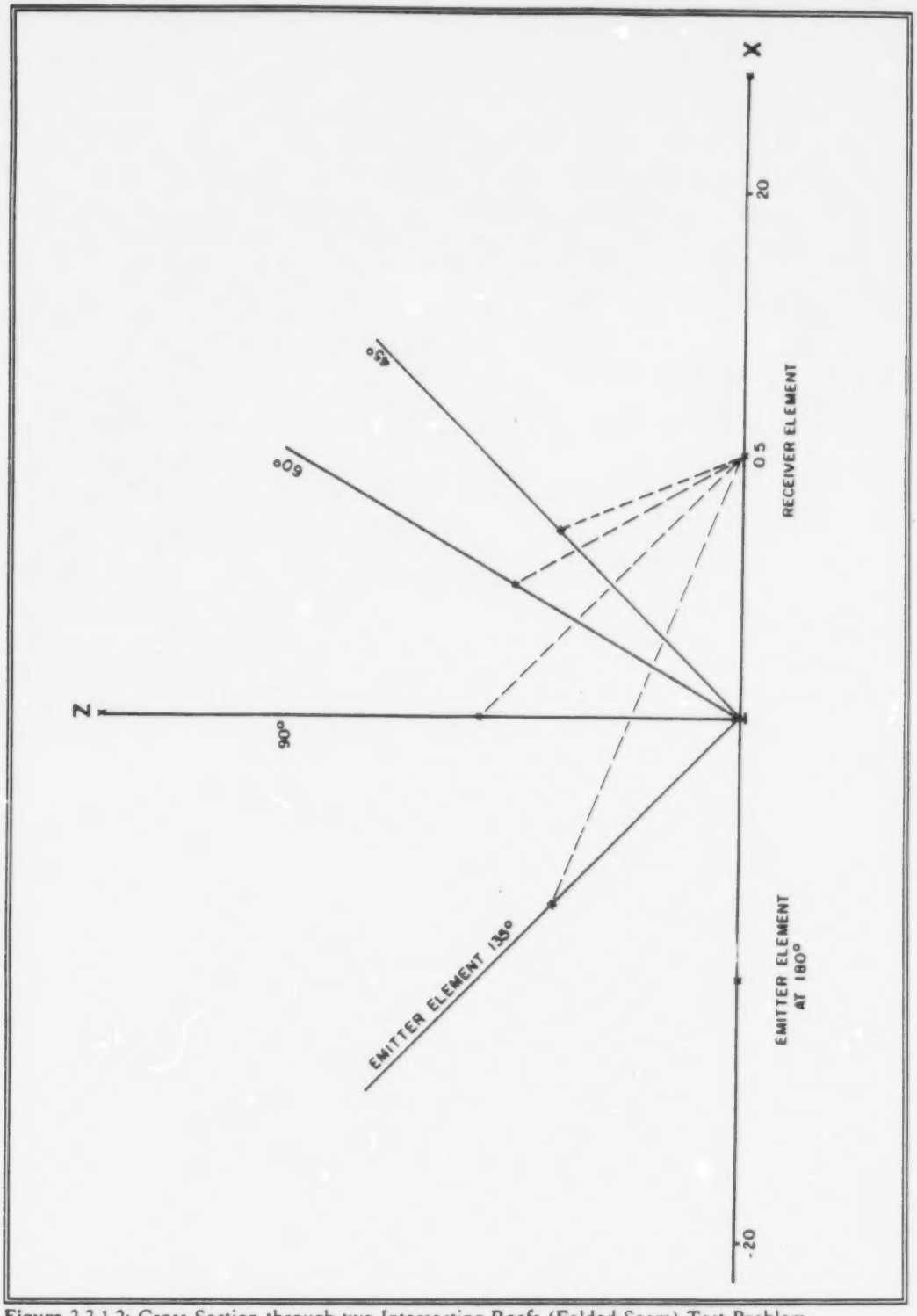


Figure 3.3.1.2: Cross Section through two Intersecting Reefs (Folded Seam) Test Problem

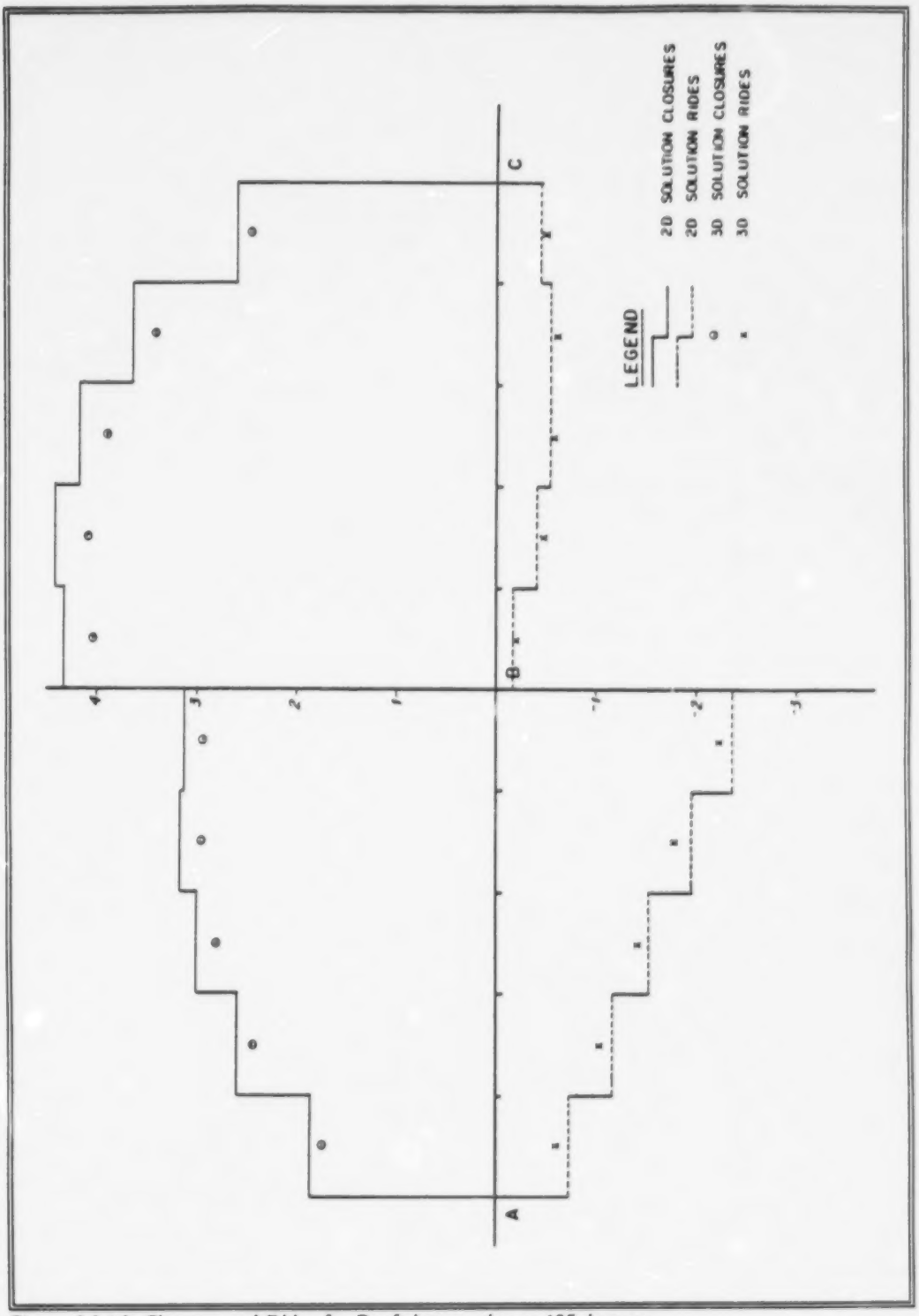


Figure 3.3.1.3: Closures and Rides for Reefs intersecting at 135 degrees

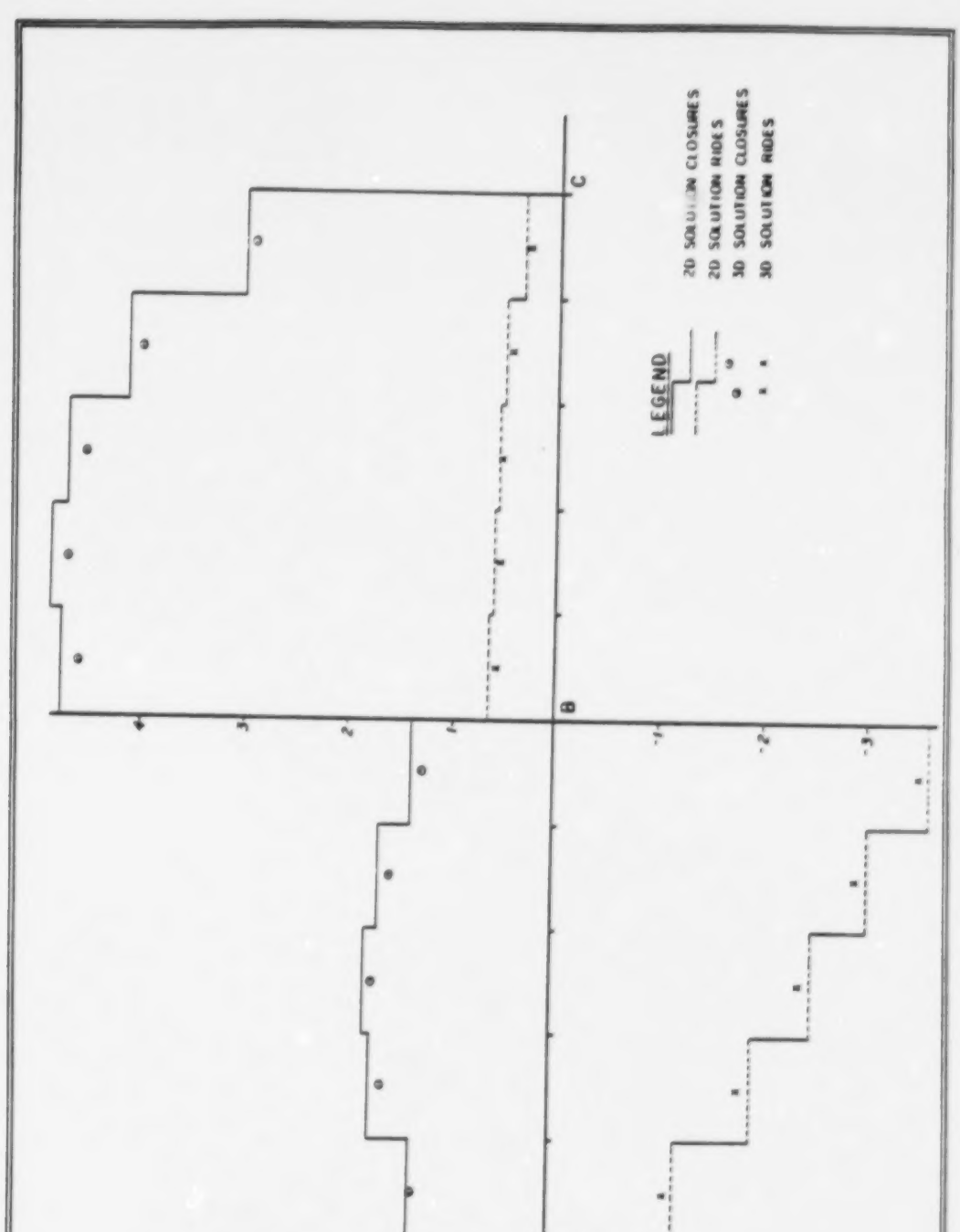


Figure 3.3.1.4: Closures and Rides for Reefs Intersecting at 90 degrees

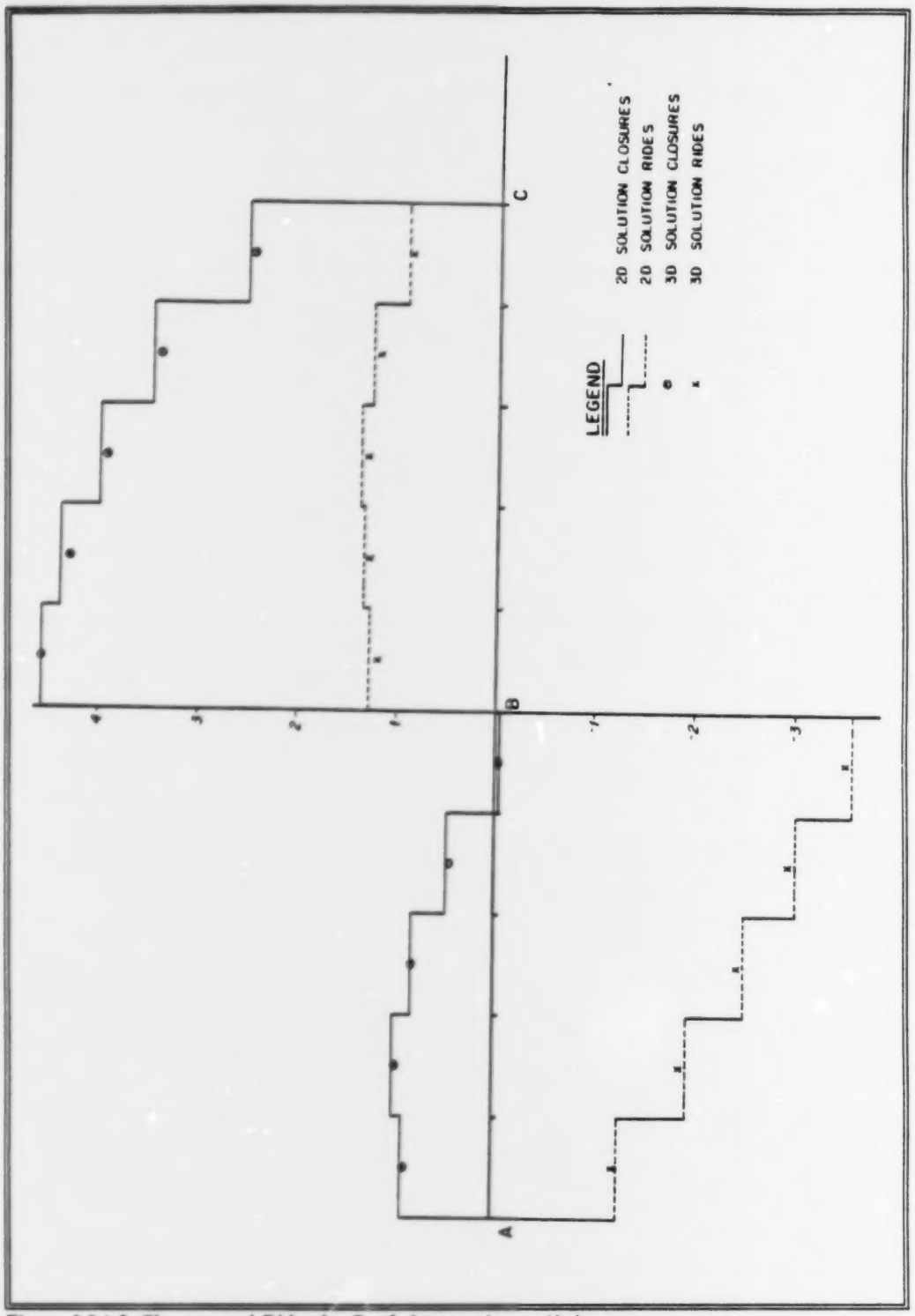


Figure 3.3.1.5: Closures and Rides for Reefs Intersecting at 60 degrees

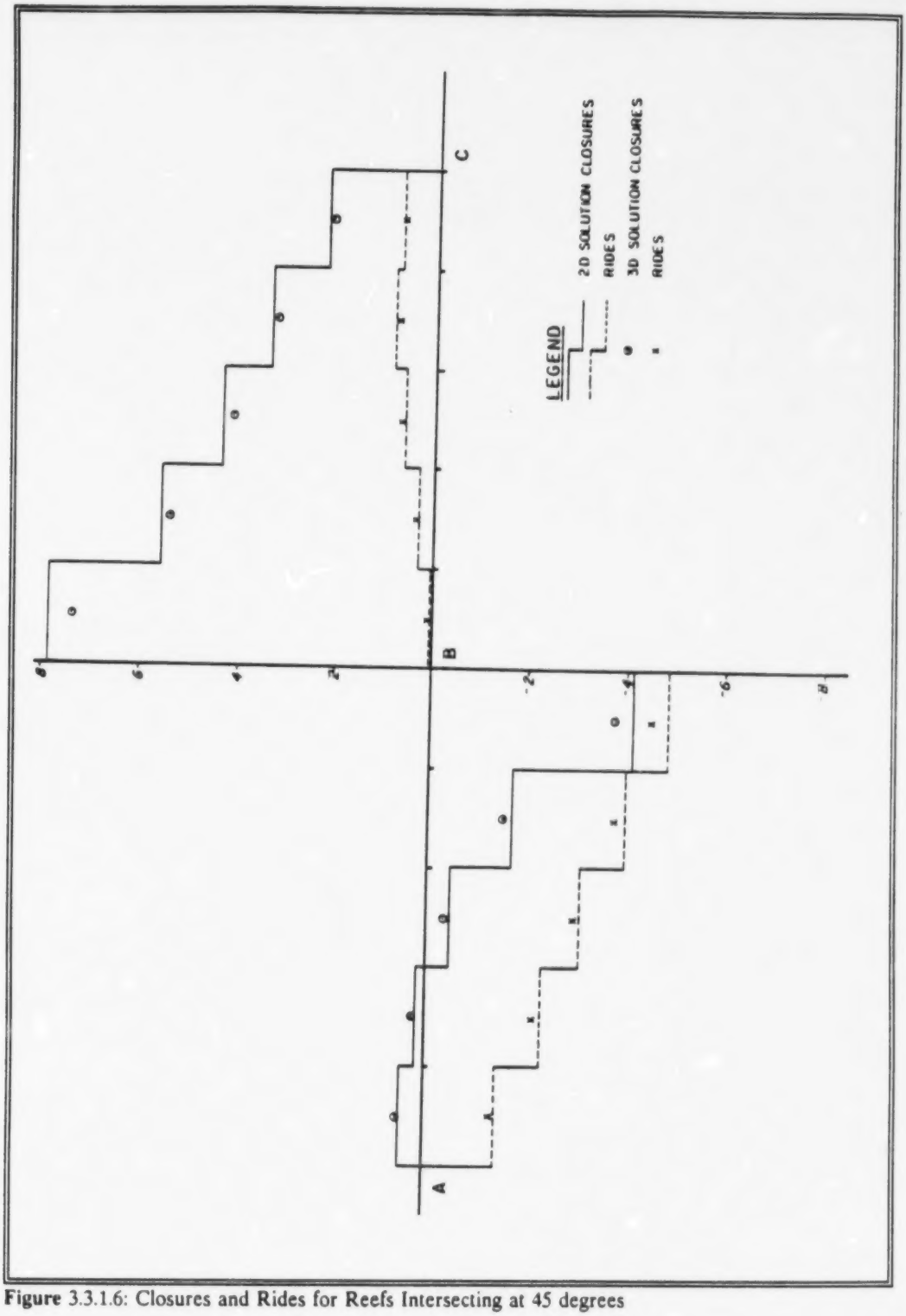


Figure 3.3.1.6: Closures and Rides for Reefs Intersecting at 45 degrees

Comparison of Coefficients for 45 Intersection Run

The coefficient describing the influence of lump 2, element 4 upon lump 1, element 3 (Fig 3.3.1.7) as calculated using an order 20 Gauss quadrature formula are:

-0.807748	-0.000640	-5.891309
0.000018	2.454829	0.000036
4.172720	-0.000011	-8.899348

and the same coefficient, when calculated using an order 10 Gauss quadrature formula are:

-0.804638	-0.000050	-5.886066
0.000010	2.458719	0.000059
4.174953	-0.0000030	-8.899068

The large coefficients are equal to within 0.5%, showing that the transition from 10 to order 20 formula at a separation ratio of 0,15 is acceptable, and also, that for lumps intersecting at angles as low as 45 can still be handled by the program, at least as regards calculation of influence coefficients. However, the coefficient (3,3) of -8.899068 is nearly as big in magnitude as the diagonal coefficient (self effect) of -12.367, showing that the diagonal dominance of the system of equations probably no longer exists and that convergence of the iterative solution will be slow (if convergence takes place at all). As has been found with other programs, if convergence occurs within a reasonable number of iterations, then the chances are very good that the solution will be good (for the choice of blocksize.

"Smoothness of Lumping"

For the test run of two reefs intersection at 60 degrees, closures and rides were plotted parallel to the line of intersection of the two reefs.

Results are shown in Figure 3.3.1.8. In this figure, results are for those elements with co-ordinates x = 19.0, z = 1.7 and y from 2 to 98. From this figure, it is evident that the effect of lumping does not produce any visible ripple effect in these curves, except possibly, at y = 20.

3.3.2 Circular excavation

This test problem was run to provide a test of the accuracy of lumping used in the program. Geometry material properties and loading are given below:

Radius of circle (R)	150	m
Youngs' Modulus E	70	GPa
Poisson's Ratio (ν)	0,2	
Element Blocksize	10	m
Primitive stress normal to excavation (P)	10	MPa

Salamon (1964) gives the following analytic solution to this problem:

$$\text{CLOSURE} = \frac{8(1-\nu^2)P}{\pi E}\left(R^2 - x^2 - y^2\right)^{1/2}$$

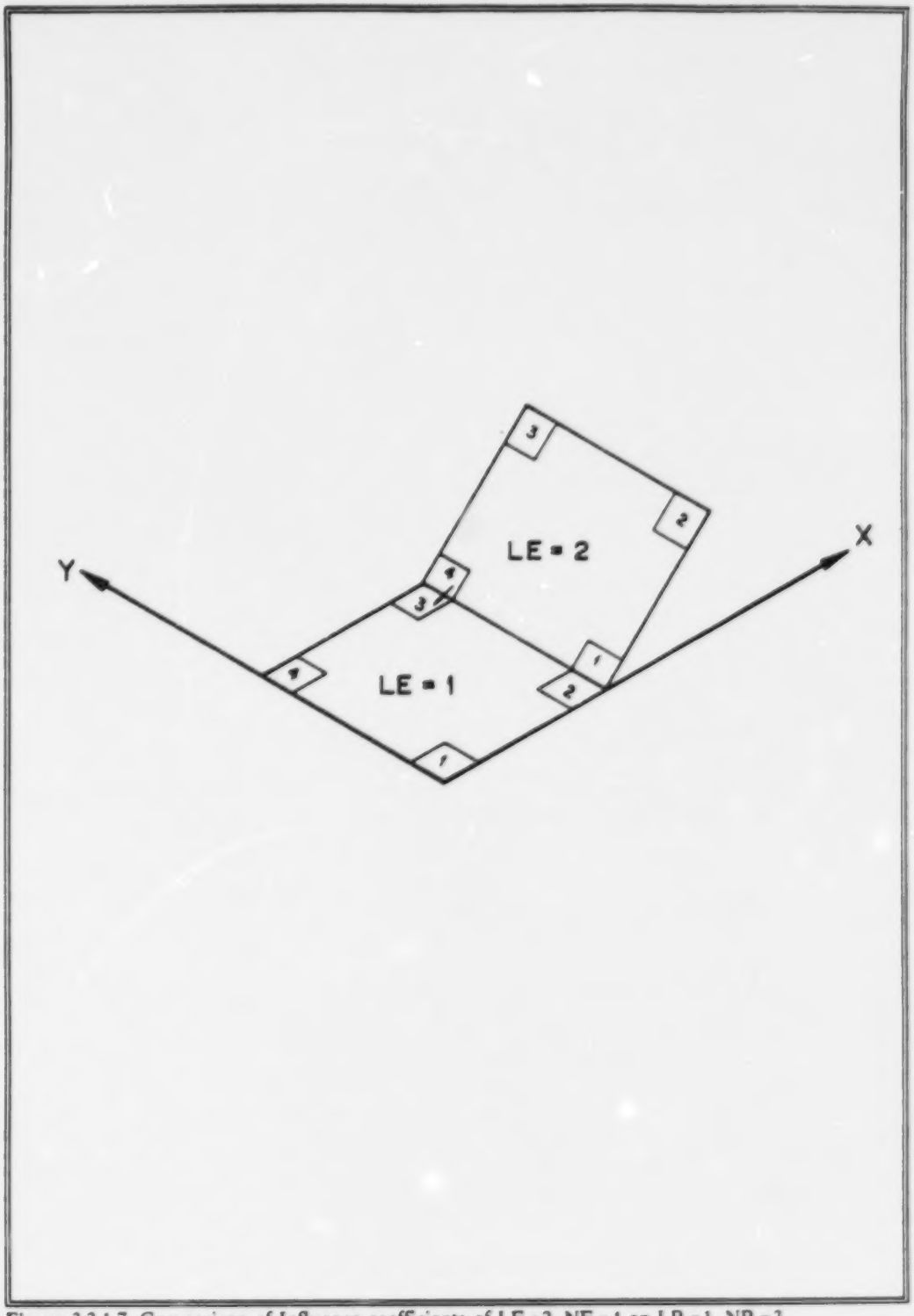


Figure 3.3.1.7: Comparison of Influence coefficients of LE = 2, NE = 4 on LR = 1, NR = 3

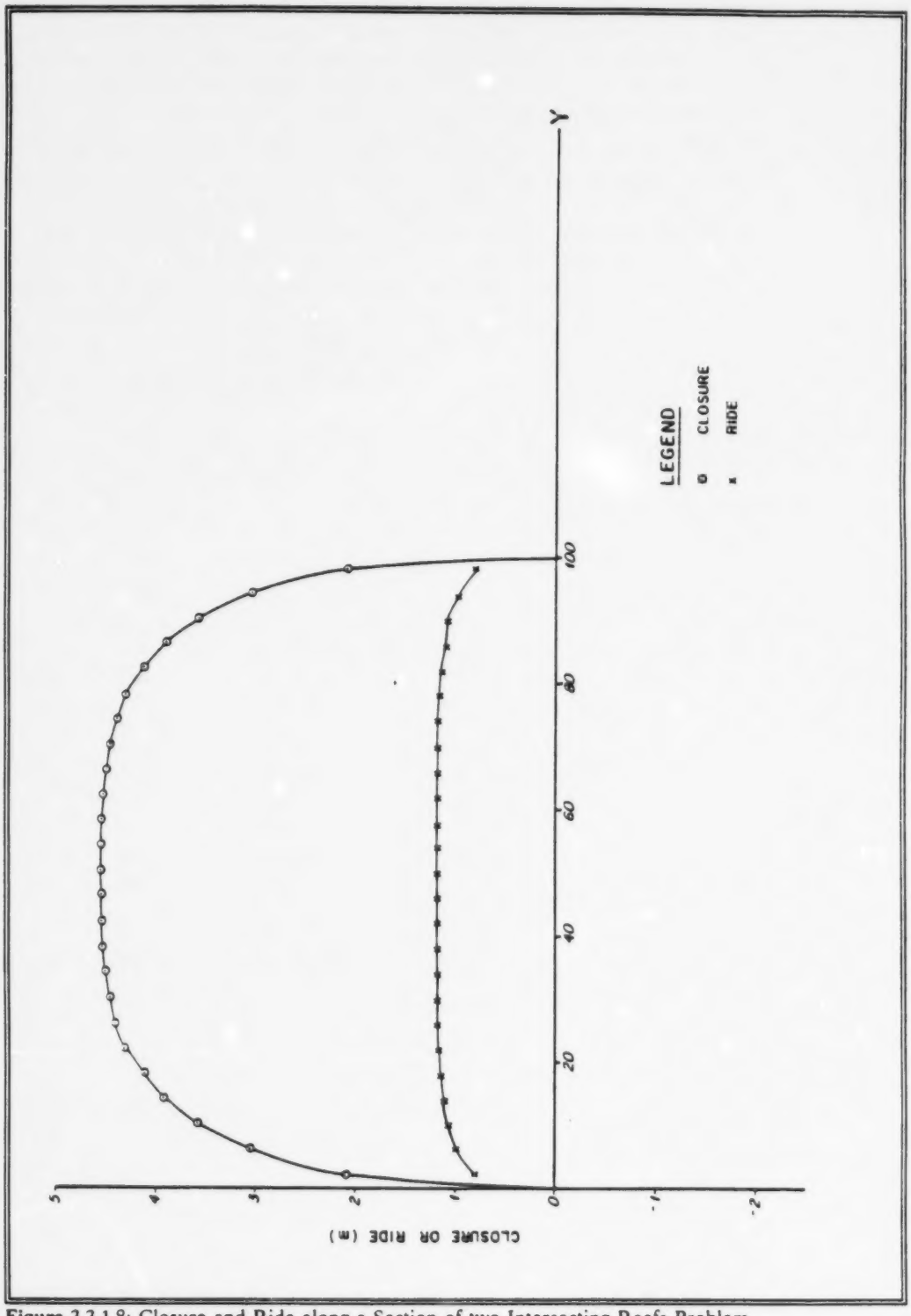


Figure 3.3.1.8: Closure and Ride along a Section of two Intersecting Reefs Problem

For the above conditions, a maximum closure of 52.4 mm is predicted. The actual curve is shown in fig. 3.3.2.1. This figure also shows actual closures calculated by BEAP-DD for a grid of 10X10 lumps. It is seen that the calculated closures oscillate around the analytic solution. This is a result of the .mping, but the magnitude of this oscillation is not considered to be excessive.

3.3.3 Shallow reef example

This example was run primarily to test the program for a large problem. 200 lumps were used to model this geometry resulting in a total of 15 000 degrees of freedom. An array of 10X10 lumps each of 100X100 metres was used to model that portion of the earth's surface above a dipping reef. An array of 10X10 lumps, each of 50X50 metres was used to model the dipping reef from a depth of 60 metres down to 200 metres.

The geometry, material properties and load conditions are summarized below:

Number of lumps:	200	
Number of elements:	5000	
Size of lumps for surface:	100	m
Size of elements for surface:	20	m
Size of lumps for reef:	50	m
Size of elements for reef:	10	m
Dip of reef:	16.26	degrees
Reef strike parallel to Y axis		
Horizontal stress x direction:	Twice vertical	
Horizontal stress y direction:	1.5 vertical	
Young's modulus, host rock:	30 000	MPa
Poisson's ratio, host rock:	0.25	
Seam thickness:	1	m
Seam modulus:	5000	MPa
Seam shear modulus:	3000	MPa

The first data file D201 which was used to set up the geometry for this problem is as follows:

```
8,101
0,0,0
1,250,250,-60
2,730,250,-200
3,730,750,-200
4,250,750,-60
5,0,0,0
6,1000,0,0
7,1000,-1000,0
8,0,1000,0
1,4,1,2,3,4
REEF PLANE
101,4,5,6,7,8
SURFACE
```

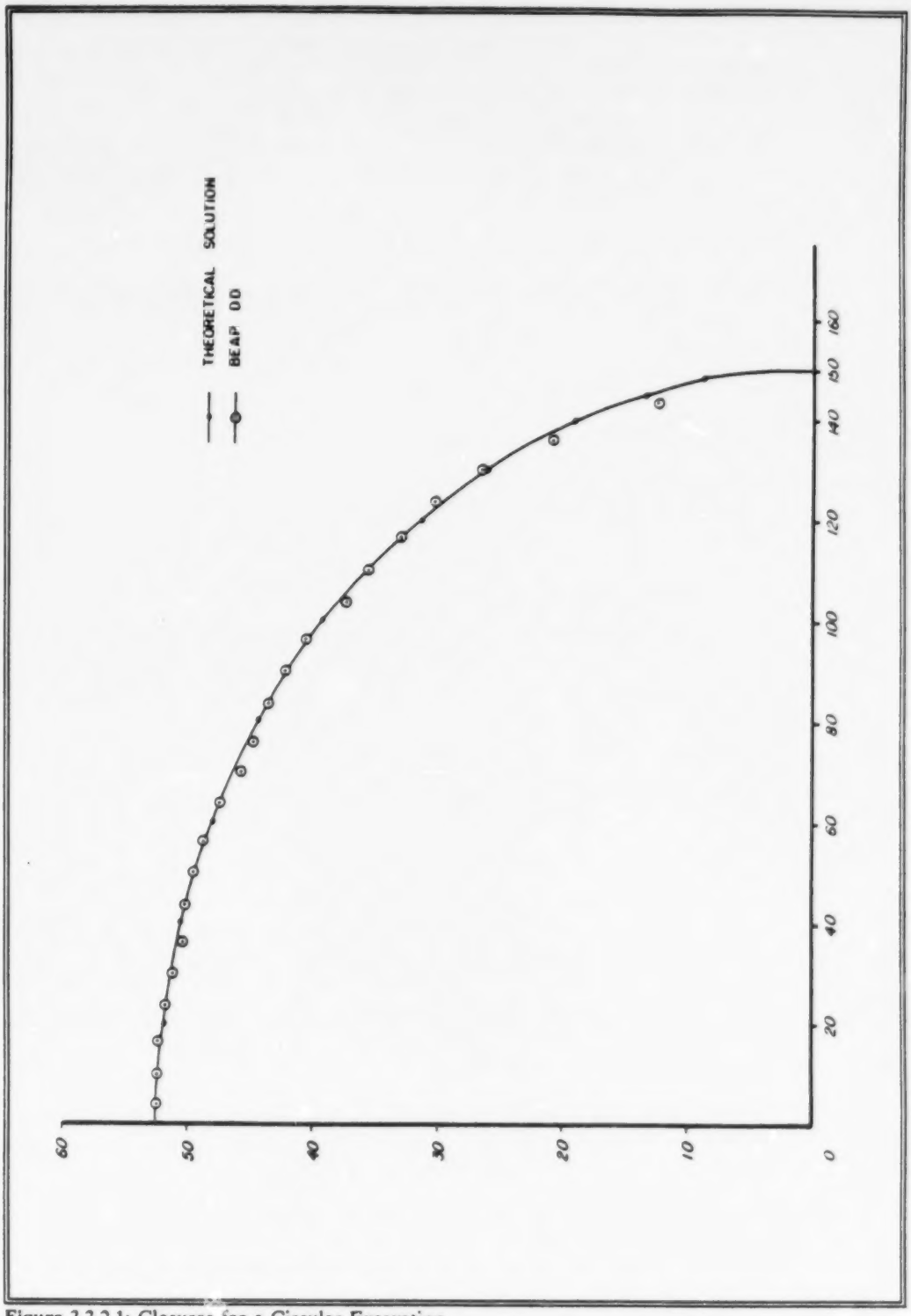


Figure 3.3.2.1: Closures for a Circular Excavation

An isometric view showing the lumps to model the earth's surface and the elements used to model the dipping reef is shown in fig. 3.3.3.1. The line of "empty squares" or lumps towards the bottom of the figure represents a portion of the reef which was assumed to be unmined.

In order for this problem to be solved, extensive use of lumping was required. In the absence of lumping, 450 megabytes of disc storage would be required to store the influence coefficients for the reef-surface and surface-reef interaction.

The following runs were carried out for comparison with one another:

Run 1 Standard 3D run with portion of earth's surface modelled using 100 lumps.

Run 2 3D run without earth's surface.

Run 3 2D run with portion of earth's surface modelled using 50 elements.

Run 4 2D run without earth's surface.

The three dimensional geometry for these runs was such that good agreement between 3D and 2D results was not expected. However, the trends observed with the inclusion or exclusion of the earth's surface should be the same for the 2D and 3D runs. In addition, the dipping reef was close enough to the earth's surface for the effect of the latter to be substantial. These effects are clearly demonstrated in the following ten figures which are summarized below.

Figure	2D/3D	Surface or Reef	Closure or Ride	Earth' surface	Curves used
3.3.3.2	3D	surface	closures	yes	1
3.3.3.3	3D	surface	rides	yes	1
3.3.3.4	3D	reef	closures	yes	1
3.3.3.5	3D	reef	rides	yes	1
3.3.3.6	3D	reef	closures	no	1
3.3.3.7	3D	reef	rides	no	1
3.3.3.8	3D	reef	closures	yes/no	2
3.3.3.9	3D	reef	rides	yes/no	2
3.3.3.10	2D+3D	reef	closures	yes/no	4
3.3.3.11	2D+3D	reef	rides	yes/no	4

These figures were prepared using the spreadsheet program Supercalc 4. The figures demonstrate the interfacing between BEAP-DD and spreadsheet programs which can then be used for the post processing.

The effect of lumping is visible in most of the curves from the 3D runs as "ripple". The magnitude of this ripple is not excessive, however, especially when it is recalled that this results from the reduction in disc storage requirements from 450 MBytes to 7.5 MBytes.

Execution time on the Data General MV 8000 for run 1 above was about 16 hours. Disk I/O for this problem is high and a substantial improvement may be achieved when running BEAP-DD on an IBM-PC/AT if use is made of virtual disk storage for certain of the data base files.

It is felt that this example, and the examples in previous sections, demonstrate that the BEAP-DD program is performing well and according to specification.

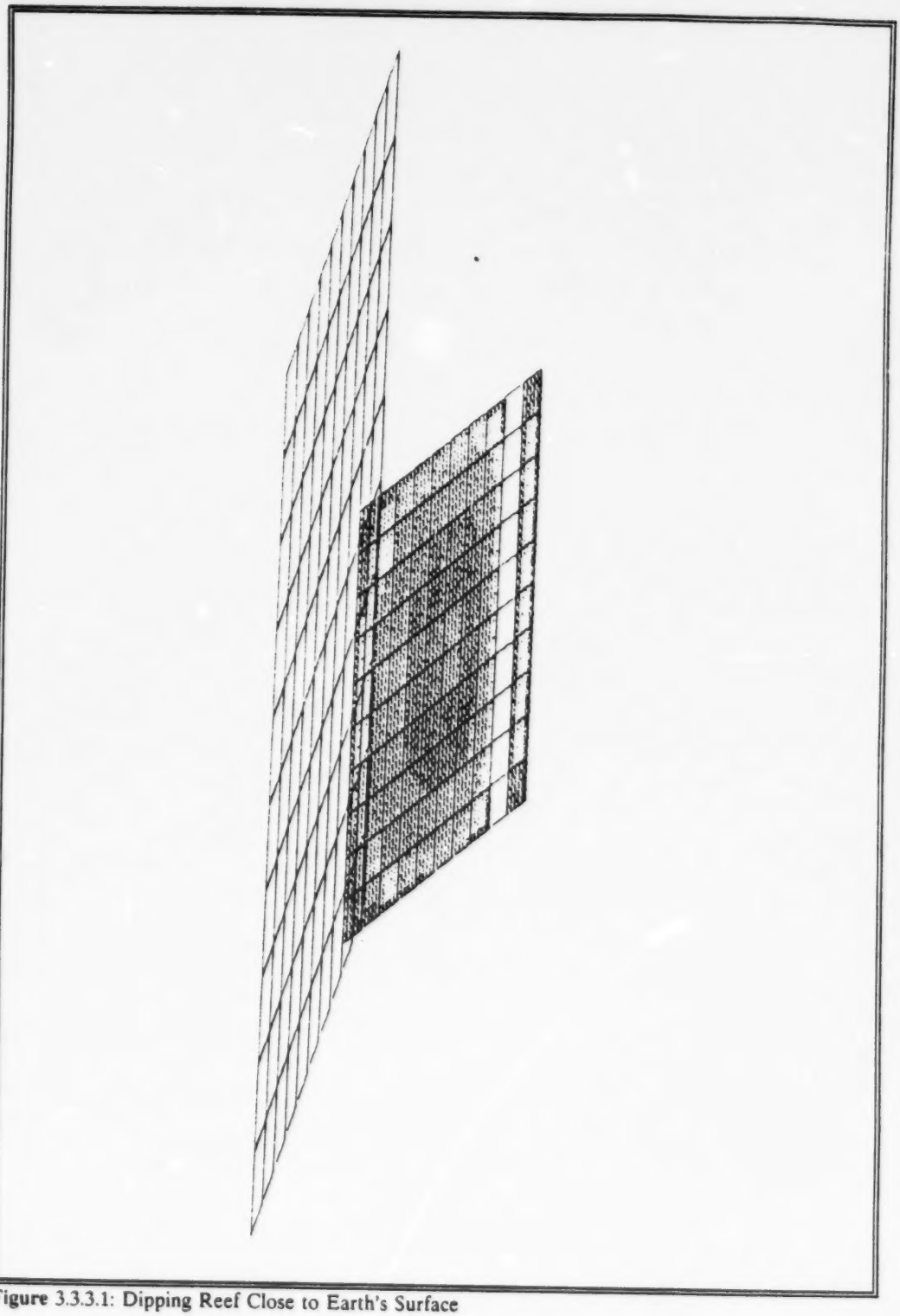


Figure 3.3.3.1: Dipping Reef Close to Earth's Surface

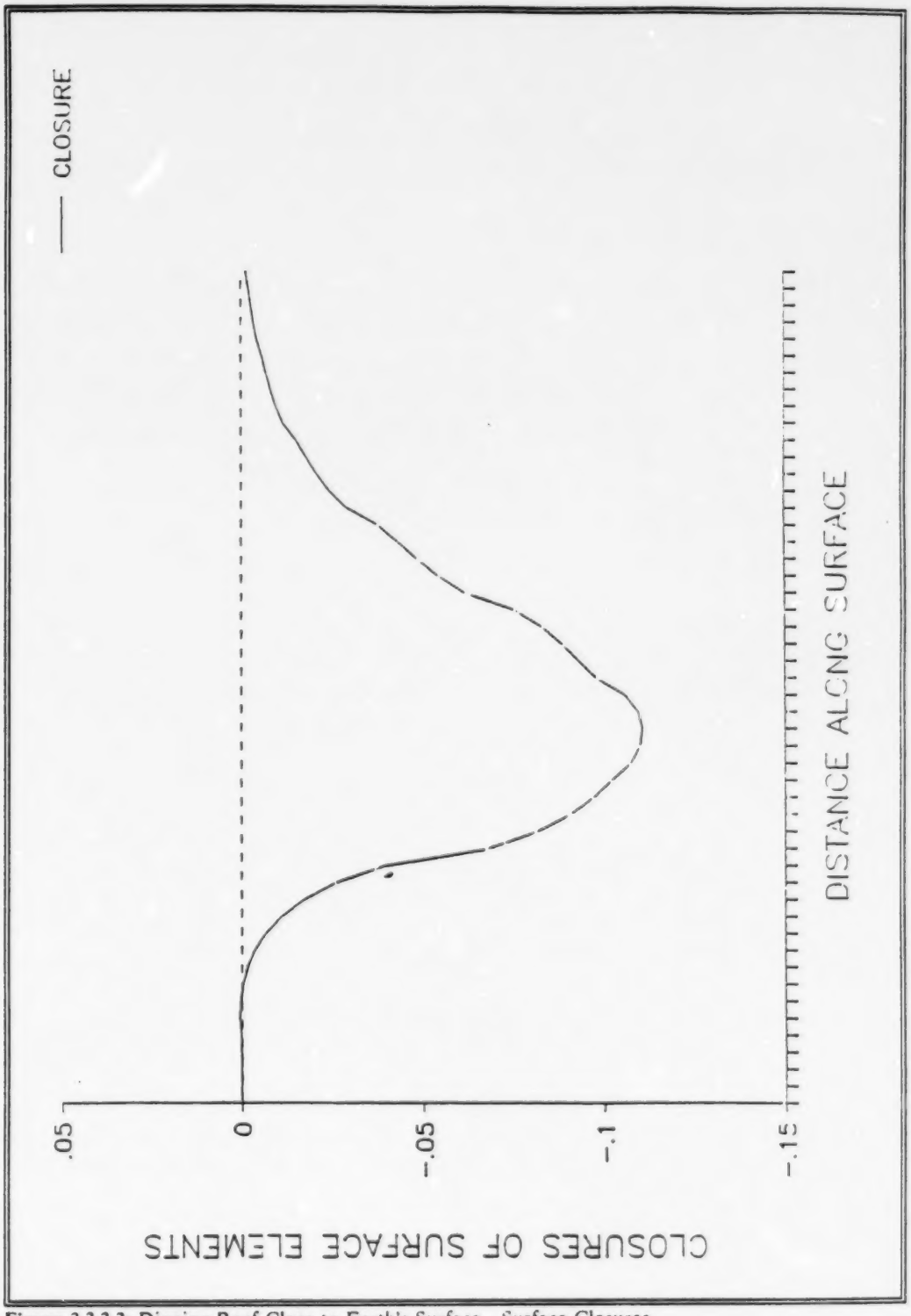


Figure 3.3.3.2: Dipping Reef Close to Earth's Surface - Surface Closures

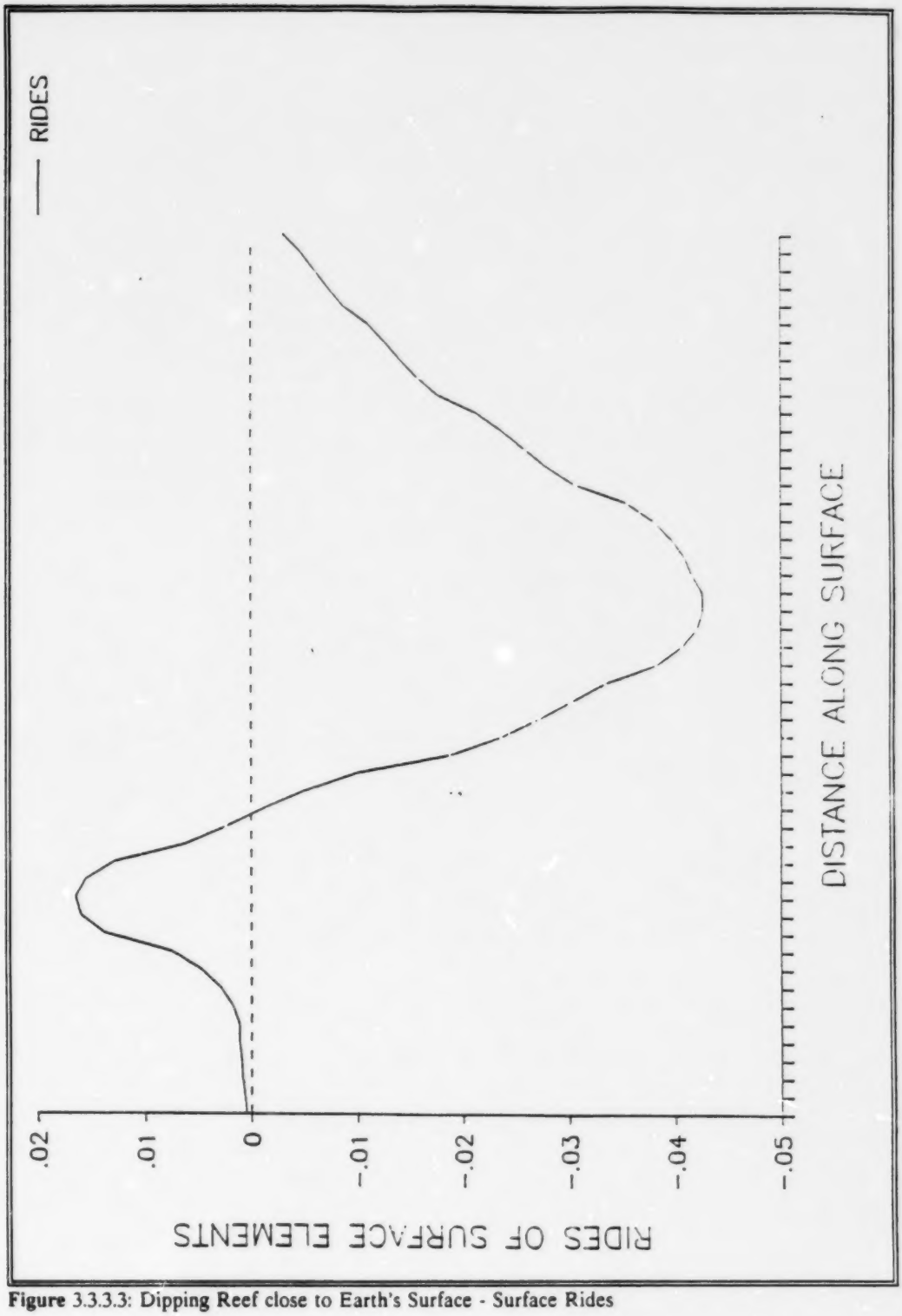


Figure 3.3.3.3: **Dipping Reef close to Earth's Surface - Surface Rides**

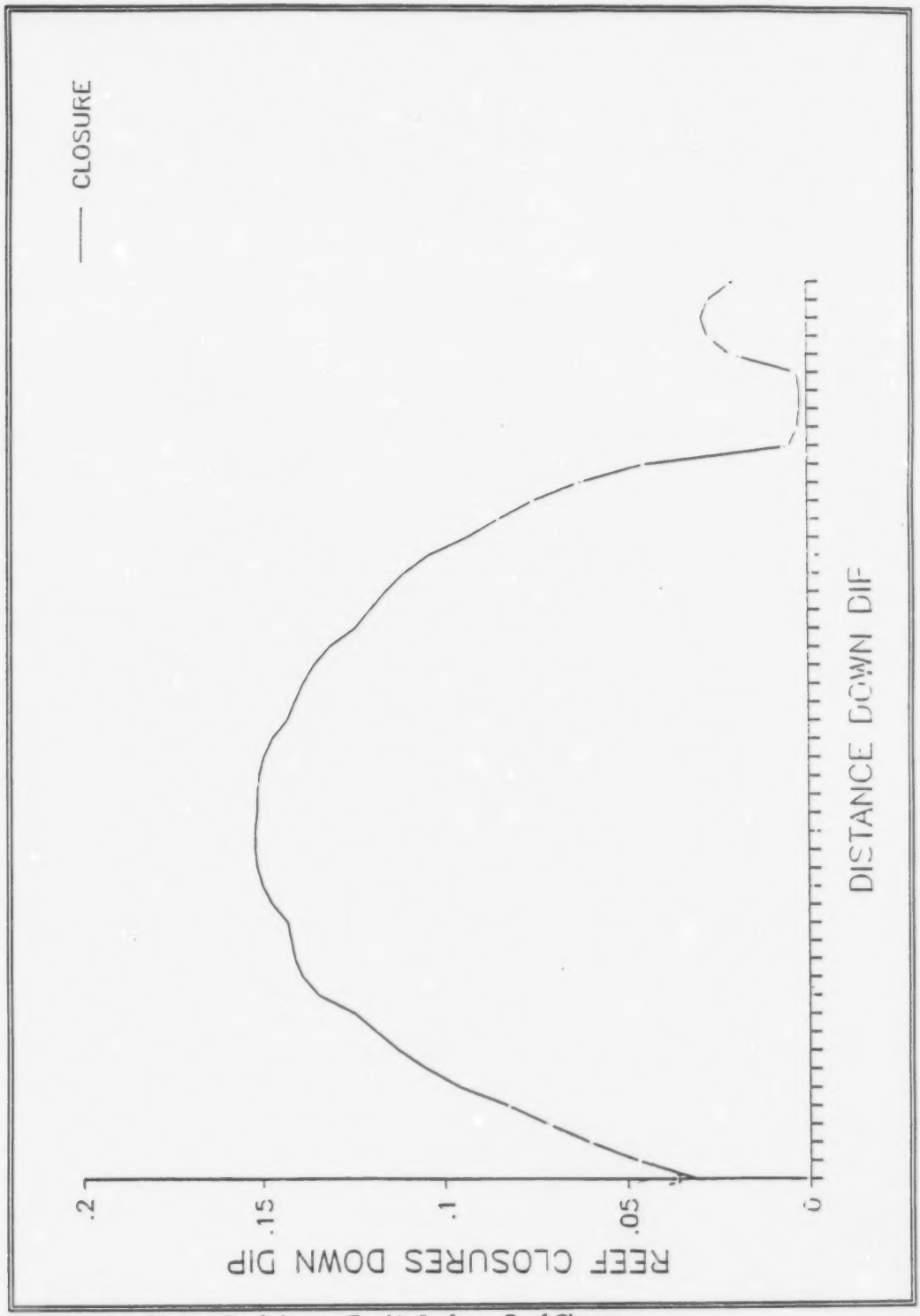


Figure 3.3.3.4: Dipping Reef close to Earth's Surface - Reef Closures

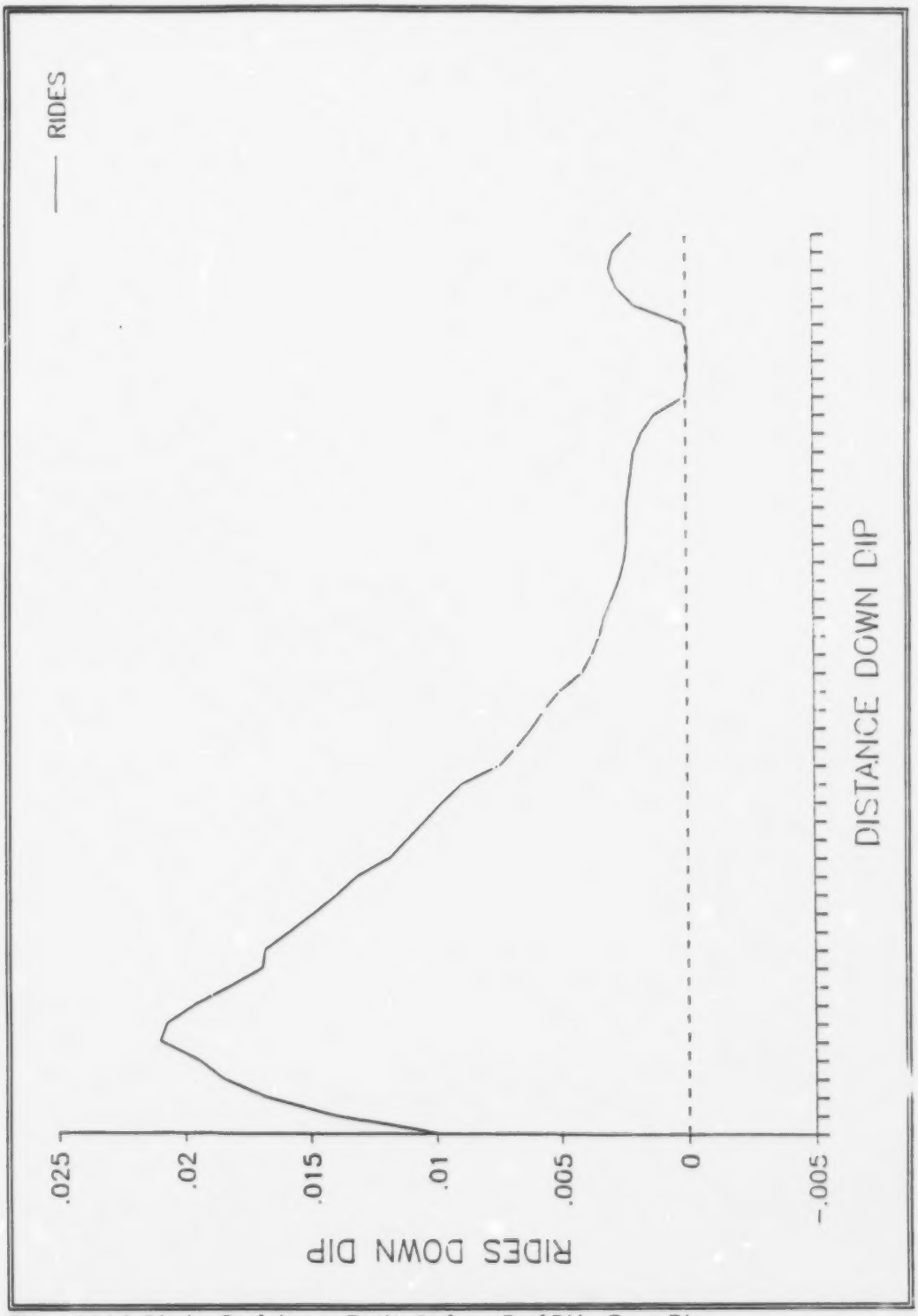


Figure 3.3.3.5: Dipping Reef close to Earth's Surface - Reef Rides Down Dip

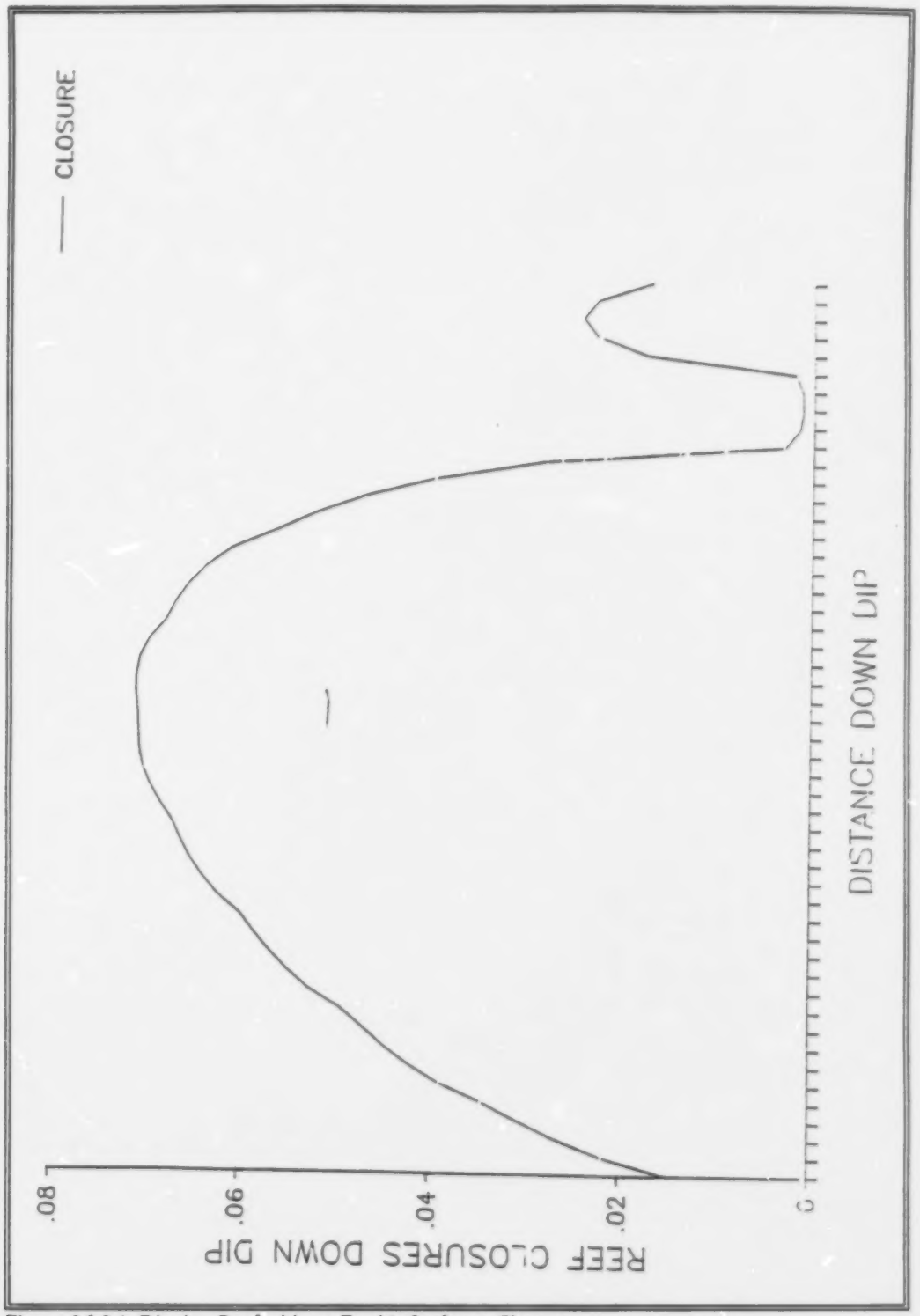


Figure 3.3.3.6: **Dipping Reef without Earth's Surface - Closures**

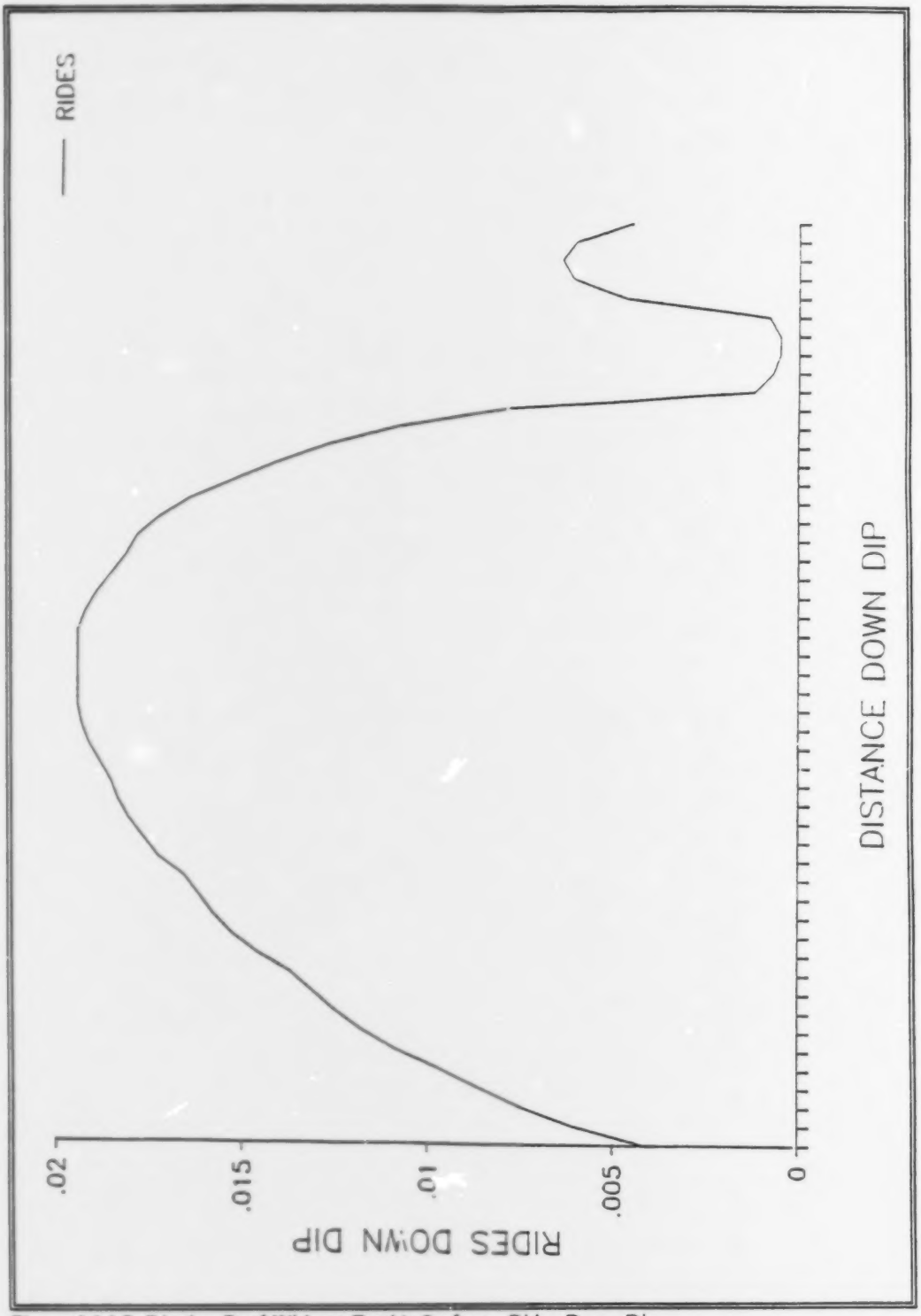


Figure 3.3.3.7: Dipping Reef Without Earth's Surface - Rides Down Dip

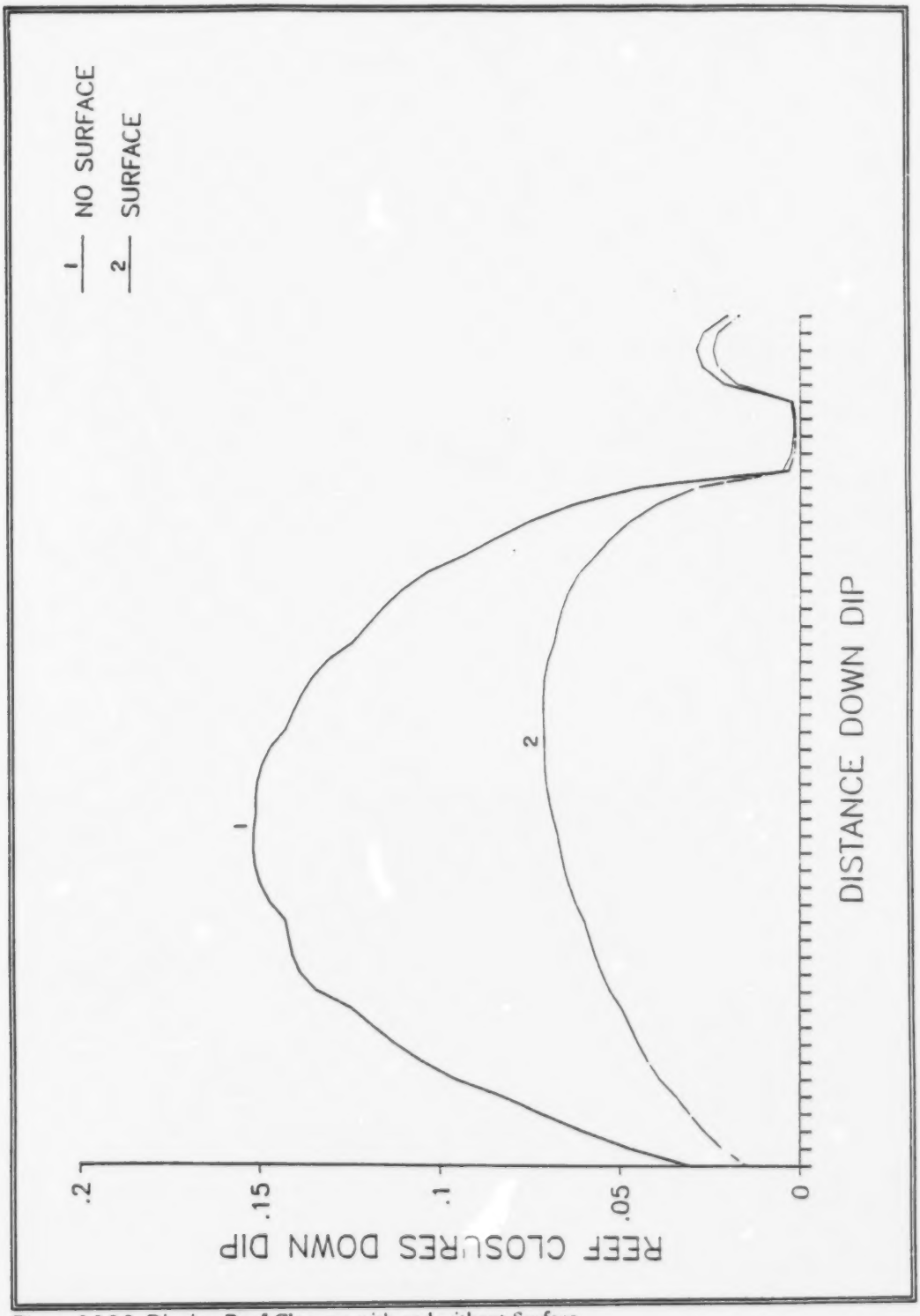


Figure 3.3.3.8: Dipping Reef Closures with and without Surface

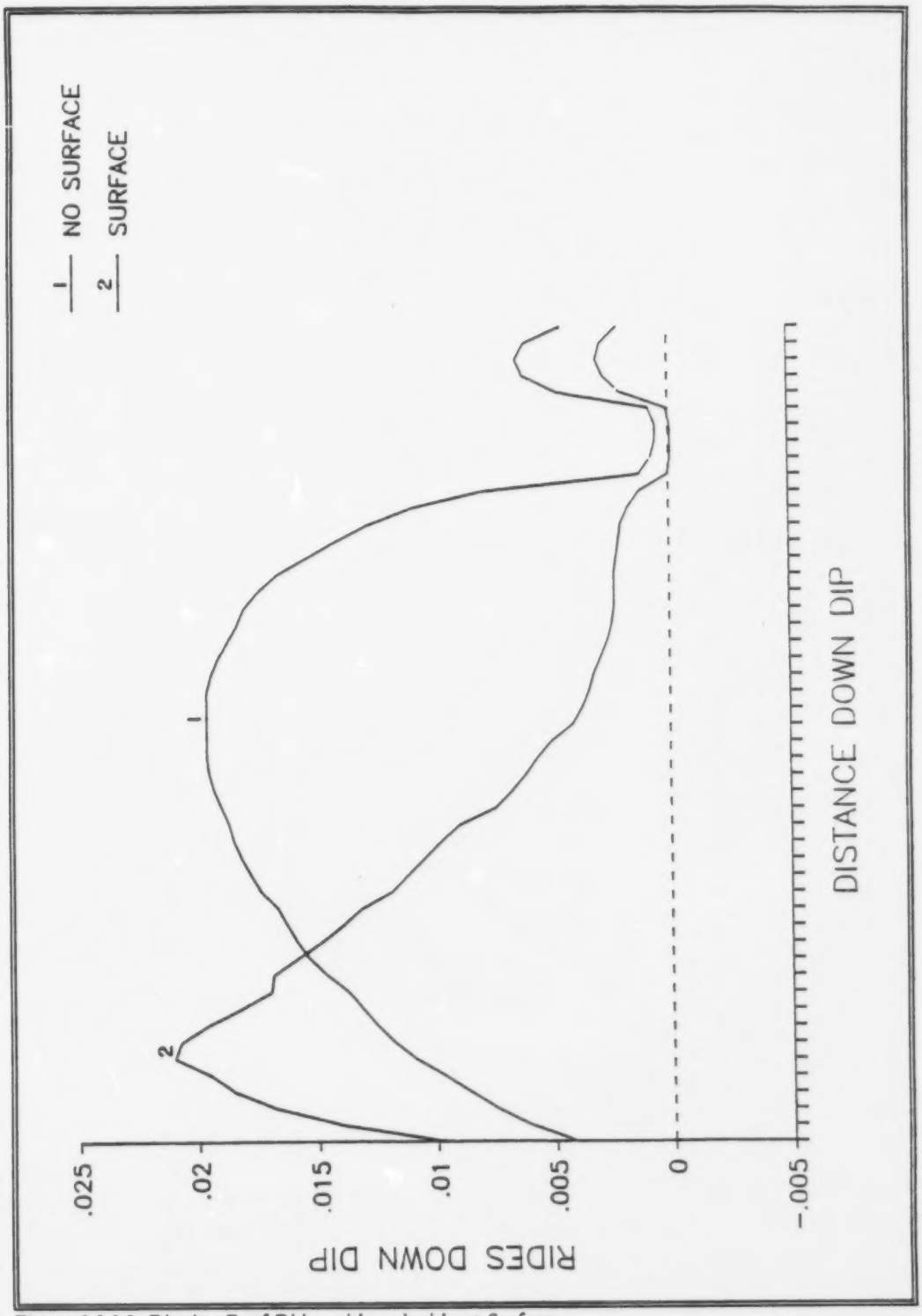


Figure 3.3.3.9: Dipping Reef Rides with and without Surface

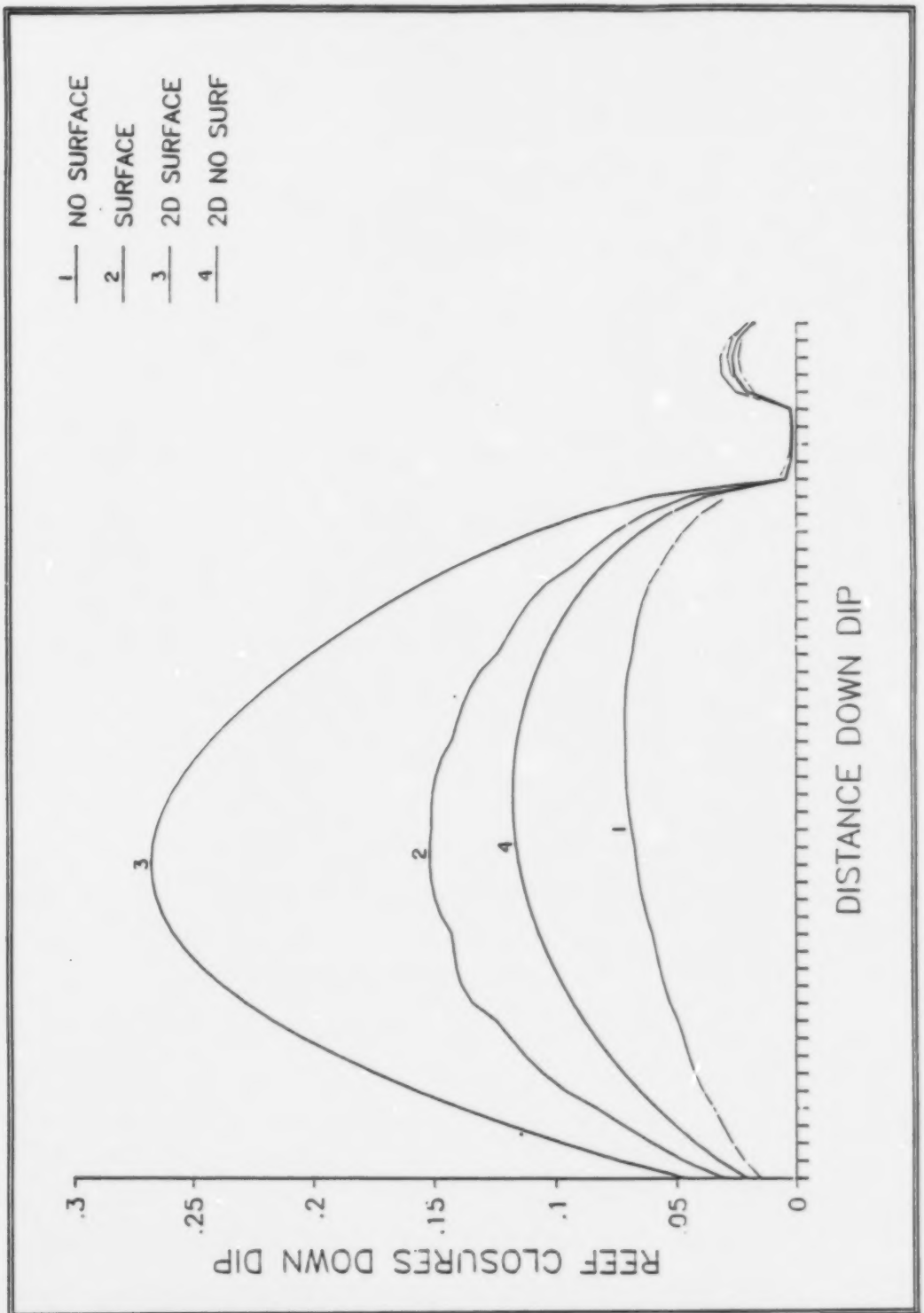


Figure 3.3.3.10: Dipping Reef with and without Earth's Surface - Comparison of 2D and 3D -Closures

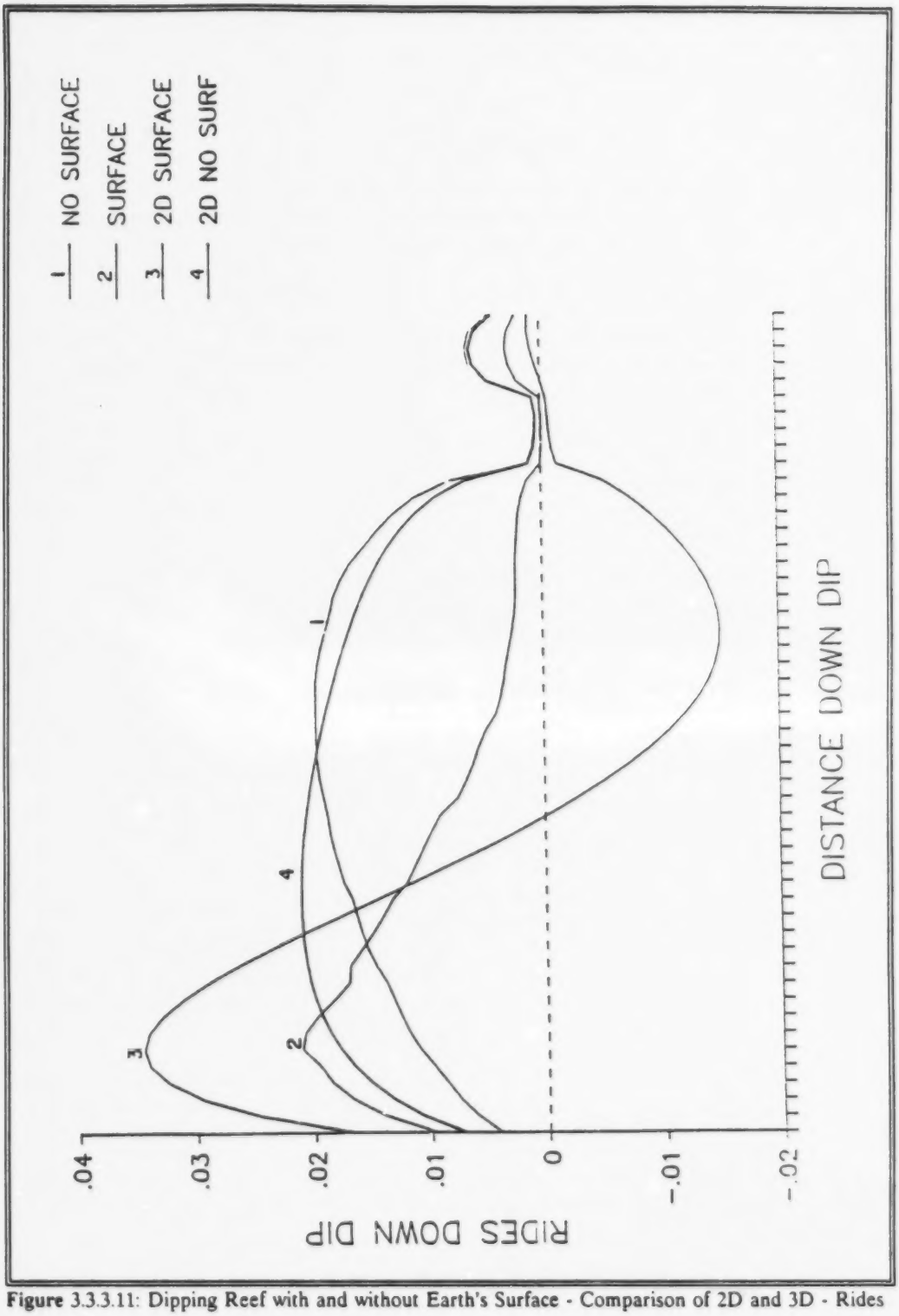


Figure 3.3.3.11: Dipping Reef with and without Earth's Surface - Comparison of 2D and 3D - Rides

3.4 BEAP-M

3.4.1 Solid cube containing a seam

The same unit cube problem has been modelled frequently with MBEM, BEAP and other boundary element codes. It is a very convenient test to establish whether the basic coding is functioning. The cube is loaded in uniaxial compression. Under this loading, the top surface of the cube will move down uniformly, and the sides will 'bulge' out uniformly according to the value of Poisson's ratio. Analytical solutions are known, so absolute checks on program accuracy are available.

In this particular test problem, however, the cube is now intersected by a displacement discontinuity grid as shown in figure 3.4.1.1. The cube is subjected to a normal loading on its upper surface (as before).

Summary information for this problem is as follows

Cube dimension	20m
Number of boundary elements	10
Number of boundary element degrees of freedom	270
Number of displacement discontinuity lumps	1
Number of displacement discontinuity elements	25
Number of displacement discontinuity degrees of freedom	75
Thickness of seam elements	1.0 m
Seam modulus (normal)	500 MPa
Seam modulus (shear)	200 MPa
Young's modulus	100 Mpa
Poisson's Ratio	0.3
Load on upper surface	-10 Mpa (down)

This is a stringent test for BEAP-M, since, no loading reaches the displacement discontinuity grid, except that calculated by the BE-DD coefficients. Similarly, if the displacement discontinuity grid closes up, then these closures will only be transmitted back to the boundary elements for the cube surfaces via the DD-BE coefficients. If the seam width is set at 1, as in this example, then if the displacement discontinuity grid is totally 'mined', then the upper portion of the cube must move down (almost as a rigid body) to accommodate this closure.

Figures 3.4.1.2 and 3.4.1.4 show vertical displacements for this problem, (which also demonstrates the multi-step modelling capability) before and after the displacement discontinuity grid is 'mined out'. Figures 3.4.1.3 and 3.4.1.5 show the stresses acting across the Seam as well as tractions on top and bottom of the cube for both simulated steps. Accuracy is within about 5%, which is acceptable, considering that the boundary elements are large compared with the displacement discontinuity elements. Also, integration accuracy could have been increased for improved accuracy.

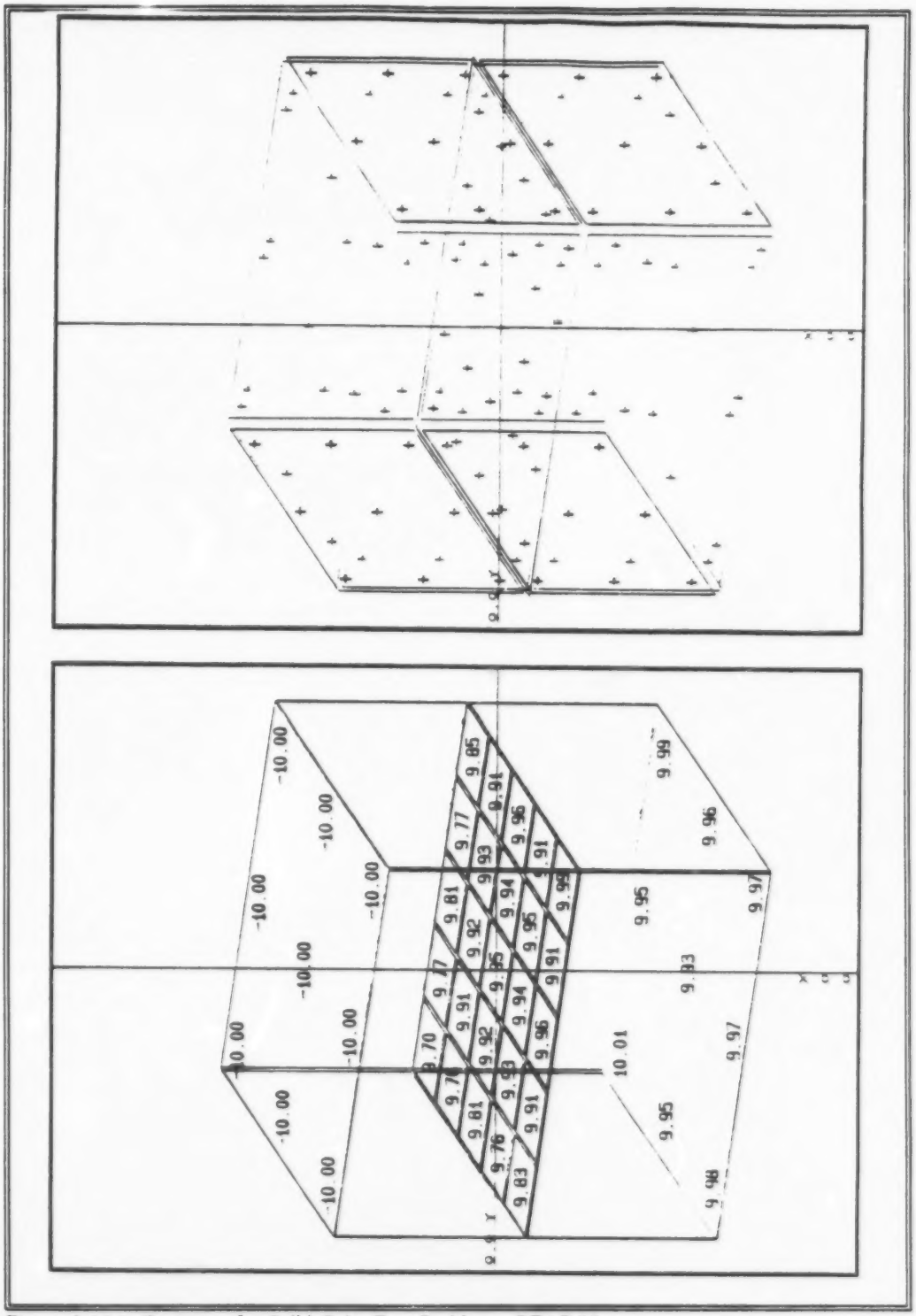


Figure 3.4.1.1: Geometry for Solid Cube Test Problem - BEAP-M

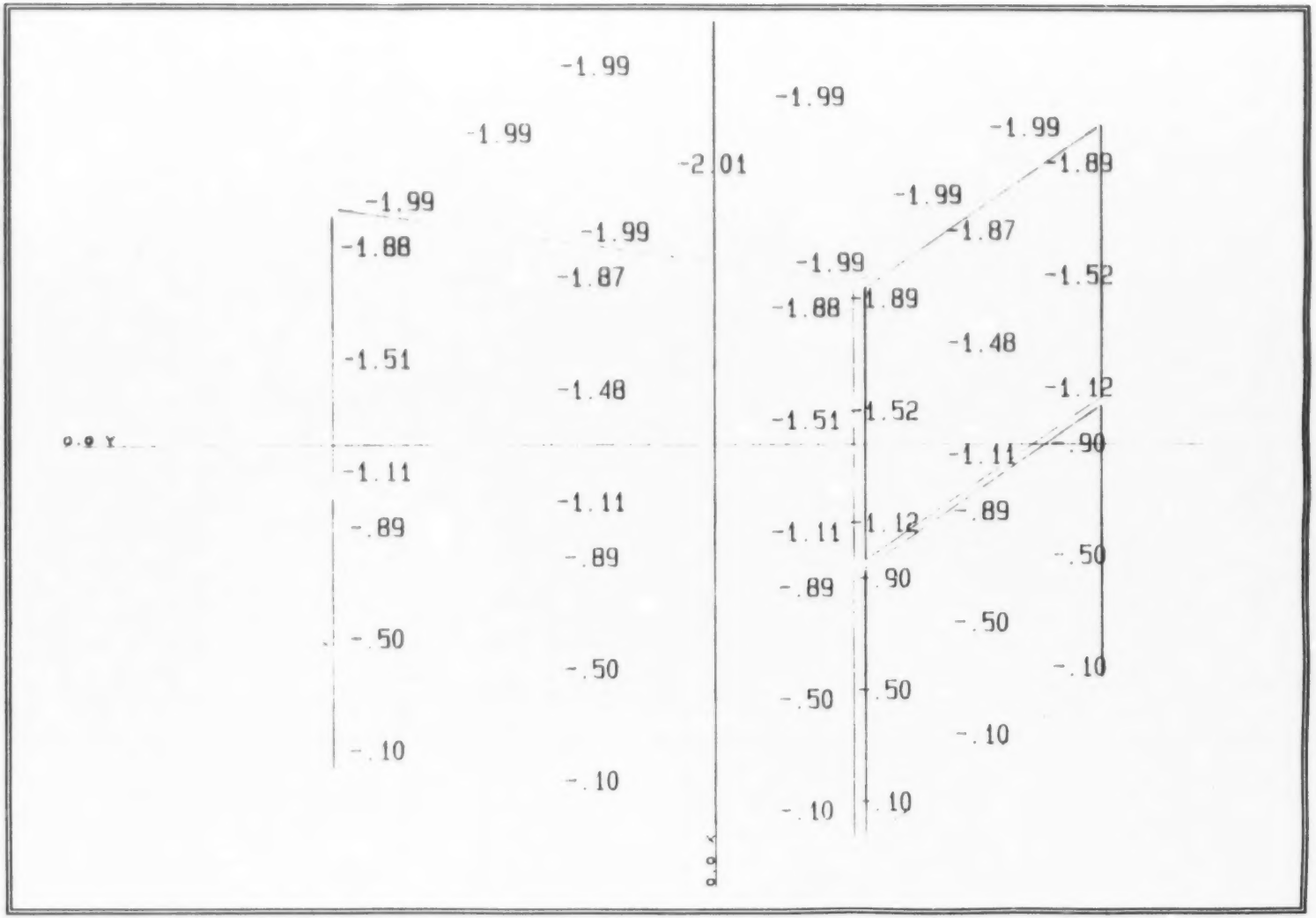


Figure 3.4.1.2: Vertical Displacements - Seam Unmined

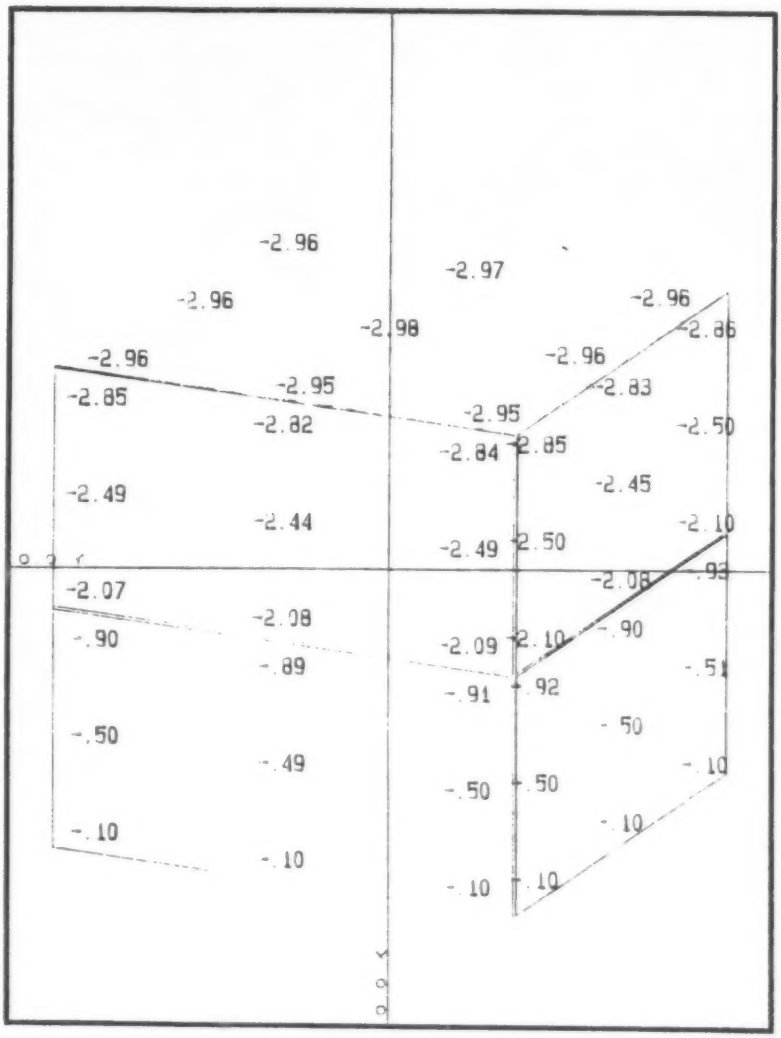
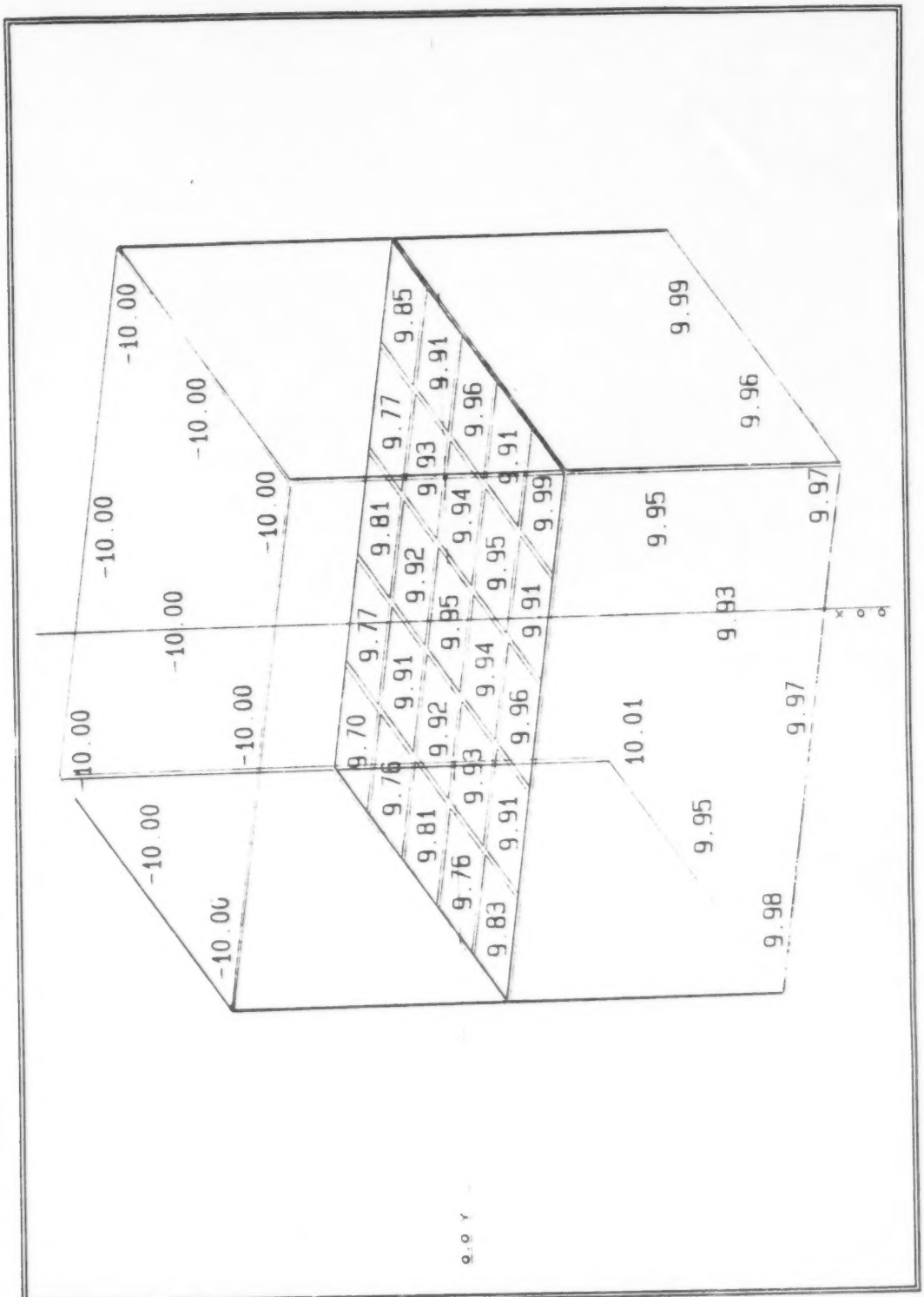


Figure 3.4.1.3: Vertical Displacement - Seam Totally Closed


-10.00
-10.00
-10.00

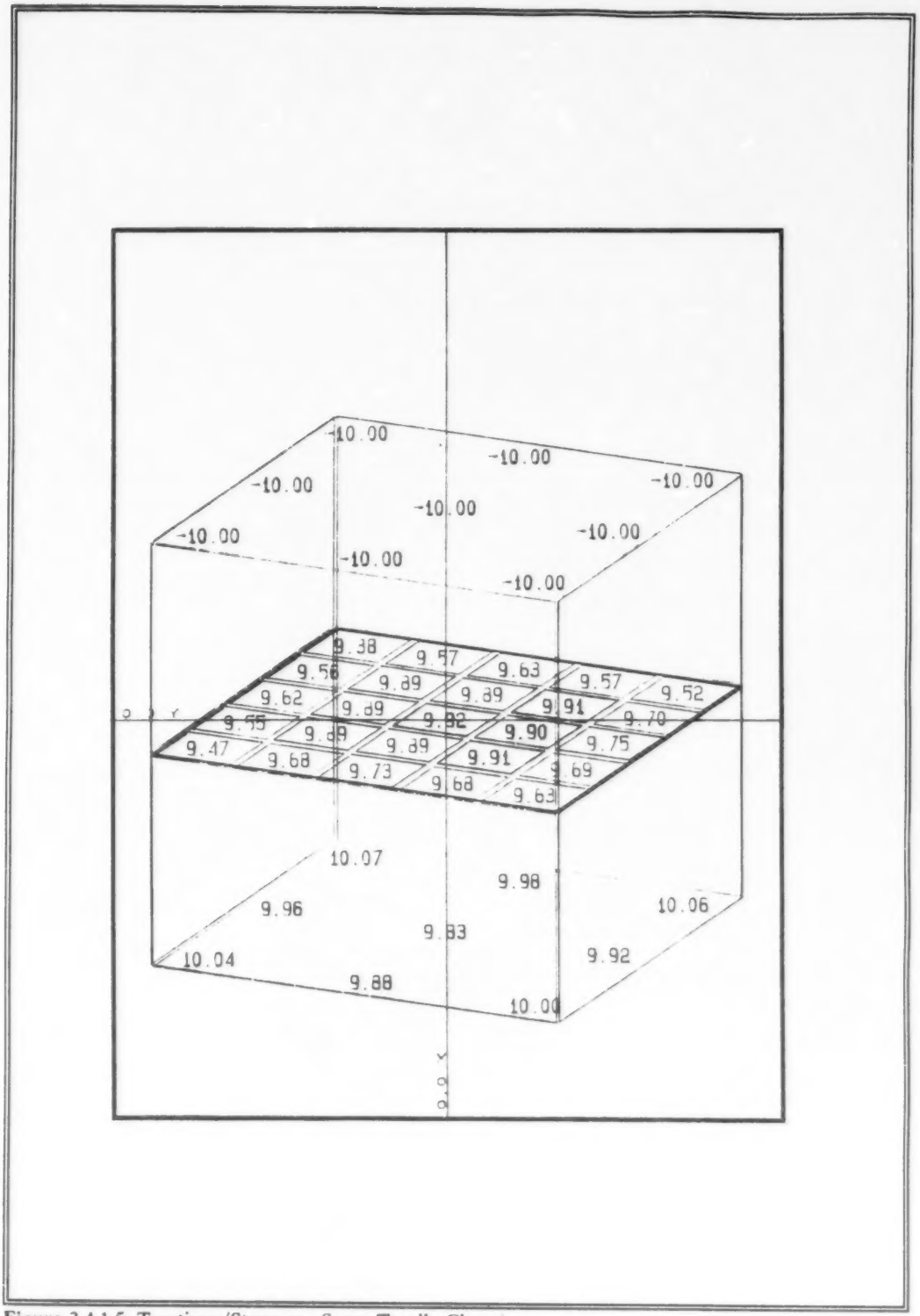


Figure 3.4.1.5: Tractions/Stresses - Seam Totally Closed

4.0 PRACTICAL EXAMPLES

4.1 UNDERGROUND EXCAVATIONS - QUAD

As the program was originally developed by Steffen Robertson and Kirsten, there have been several opportunities to use the program for practical (real world) problems. An example of a mesh for such a problem is given in figure 4.1.1. More examples are included with the QUAD user documentation.

4.2 FIVE INTERACTING EXCAVATIONS BEAP

This problem is included here since it has been modelled both using BEAP and the older MBEM program. It has also been studied using other three-dimensional formulations and has been reported in various papers.

The problem to be described here is that described by Cowling (1985) and Watson and Cowling (1985). It considers five excavations in close proximity to one another, taken to represent a portion of the primary stoping of the Mount Isa 1100 orebody. This problem has been studied using the three-dimensional program THREE (Watson and Cowling (1985)) and subsequently using programs MBEM, Diering (1982) and BEAP in this study. THREE uses conforming quadratic variation elements MBEM uses constant displacement/traction elements, whilst BEAP, as described in the previous sections, uses non-conforming quadratic variation elements. Detailed geometry, loading and material properties as supplied by Cowling (1985) were used here and loading and material properties are summarized below.

1. Rock Properties

Silica Dolomite Young's modulus 80 GPA Poisson's ratio 0,25

2. Coordinate system

	x1	East-west
	x2	North-south
	x3	Vertical (+ upwards)

3. Stress field

Load condition 1

Stress	E - W	= 53.6 MPa - 0.0080 MPa * x3
Stress	N - S	= 60.0 MPa - 0.0152 MPa * x3
Stress	Vertical	= 95.5 MPa - 0.0274 MPa * x3
Shear	E-W/N-S	= -7.5 MPa

Load condition 2

Stress	E - W	= 99.7 MPa - 0.026 MPa *x3
Stress	N - S	= 48.4 MPa - 0.012 Mpa *x3
Stress	Vertical	= 96.6 MPa - 0.028 MPa *x3
Shear	E-W/Vertical	= 5.15 MPa

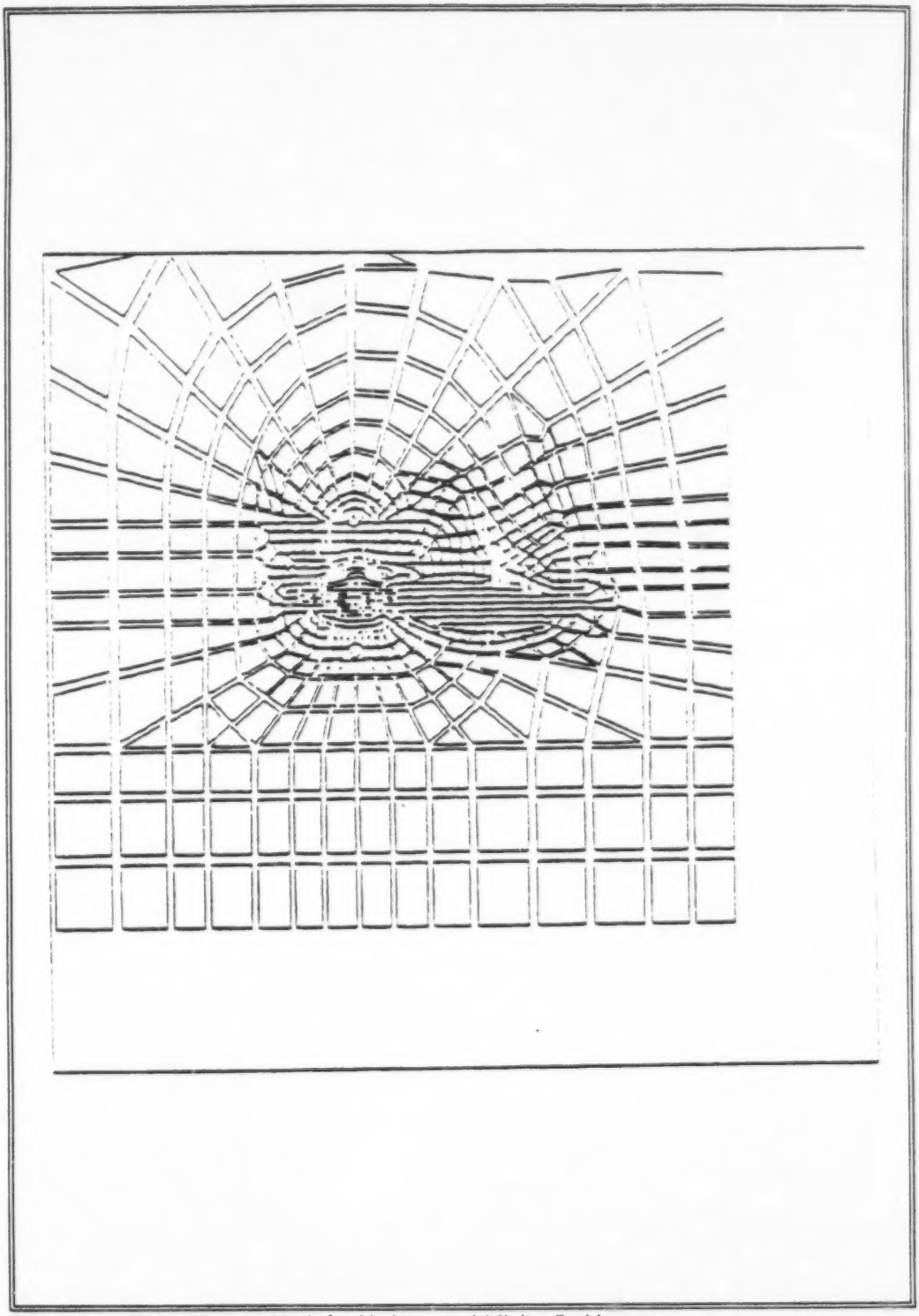


Figure 4.1.1: Finite Element Mesh for Underground Mining Problem

Two isometric views of the geometry (looking down and looking up) are shown in Figures 4.2.1 and 4.2.2 respectively. Various runs have been carried out for this study, and these are compared with results obtained from Cowling (1985) which are essentially the same as those published by Watson and Cowling (1985).

Program	Run	Load Condition	Ele-ments	Reference
THREE	1	2	46	Cowling (1985), Watson & Cowling(1985)
	2	2	136	Cowling (1985), Watson & Cowling(1985)
	3	2	207	Cowling (1985), Watson & Cowling(1985)
MBEM	5	2	470	This study
BEAP	1	2	46	This study
	2	2	184	This study
BEAP	3	1	46	This study
	4	1	184	This study
MBEM	1	1	414	This study
	2	1	736	This study
	3	1	953	This study
	4	1	2304	This study

For these runs, stresses are compared at two points A and B, the locations of which are shown schematically in Figure 4.2.3. The points lie approximately at the centres of the pillars formed by the two lines of stopes.

Results at these two points A and B are summarized below for the two load cases.

		Principle Stresses A			Principle Stresses B		
Program	Run	Min	Int	Max	Min	Int	Max
THREE	1	8.2	25.1	42.5	7.9	25.6	44.5
THREE	2	8.8	24.8	42.9	3.5	25.7	45.5
	3	5.3	24.9	43.3	0.5	25.7	46.7
MBEM	5	-1.6	22.8	42.7	-0.6	24.4	44.5
BEAP	1	-0.1	23.6	42.7	1.2	24.7	44.5
BEAP	2	-0.4	23.6	42.9	0.9	24.9	44.7
Measured		13.8	20.3	32.0	13.8	19.2	40.9
Primitive (pre-mining)		17.2	21.3	34.5	17.5	22.0	35.2

TABLE 4.2.1 PRINCIPAL STRESSES AT A AND B FOR LOAD CASE 1

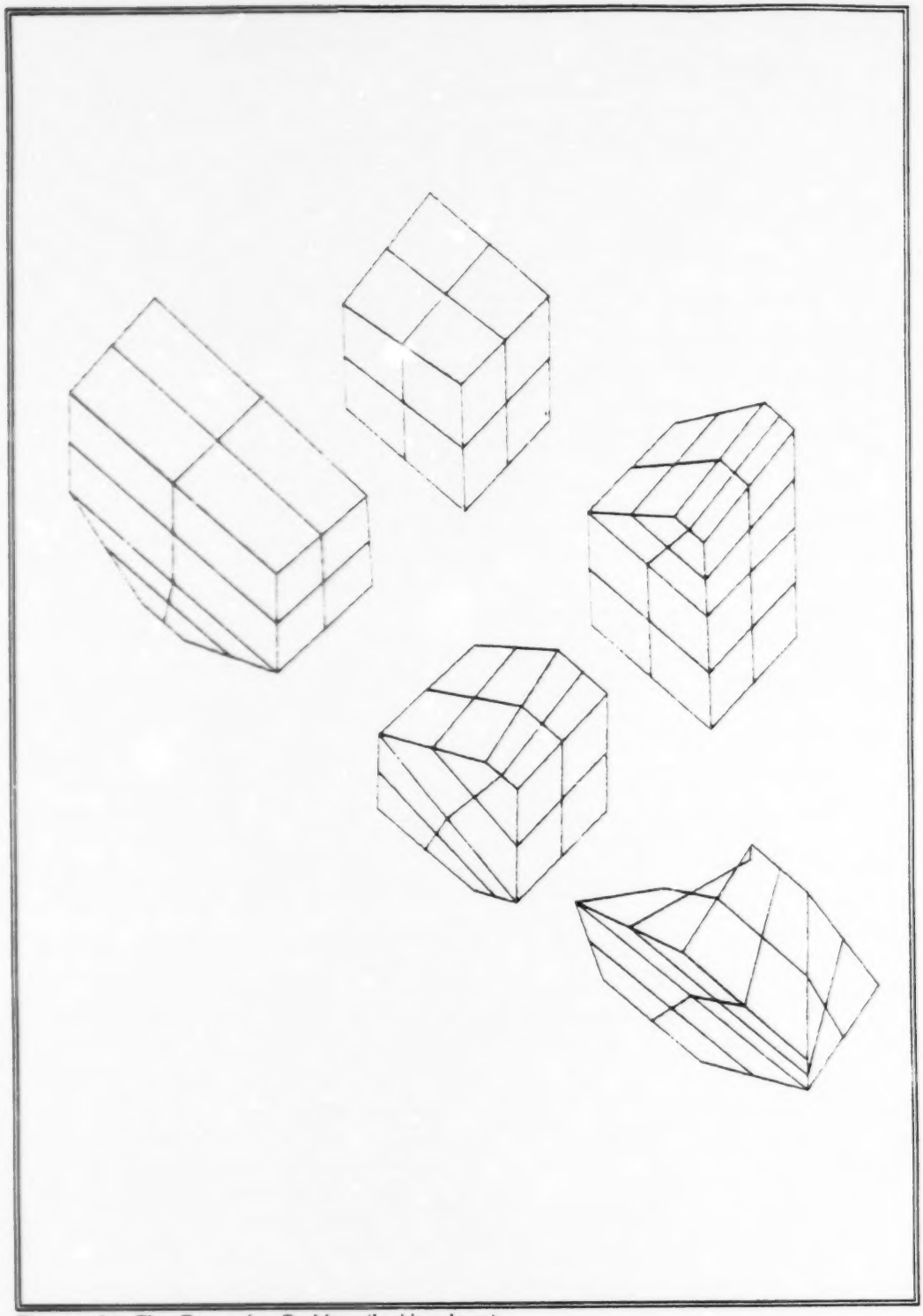


Figure 4.2.1: Five Excavation Problem (looking down)

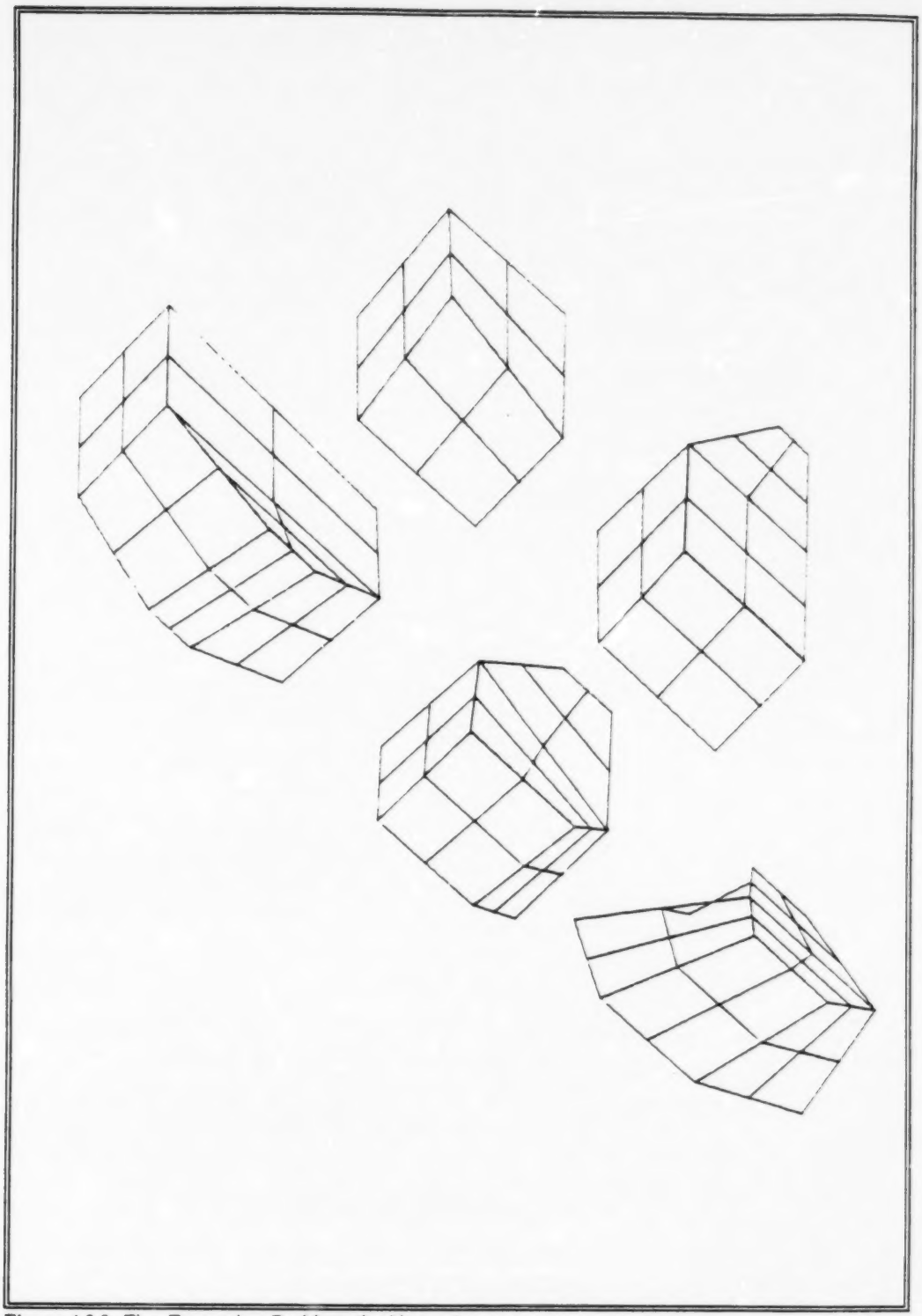


Figure 4.2.2: Five Excavation Problem (looking up)

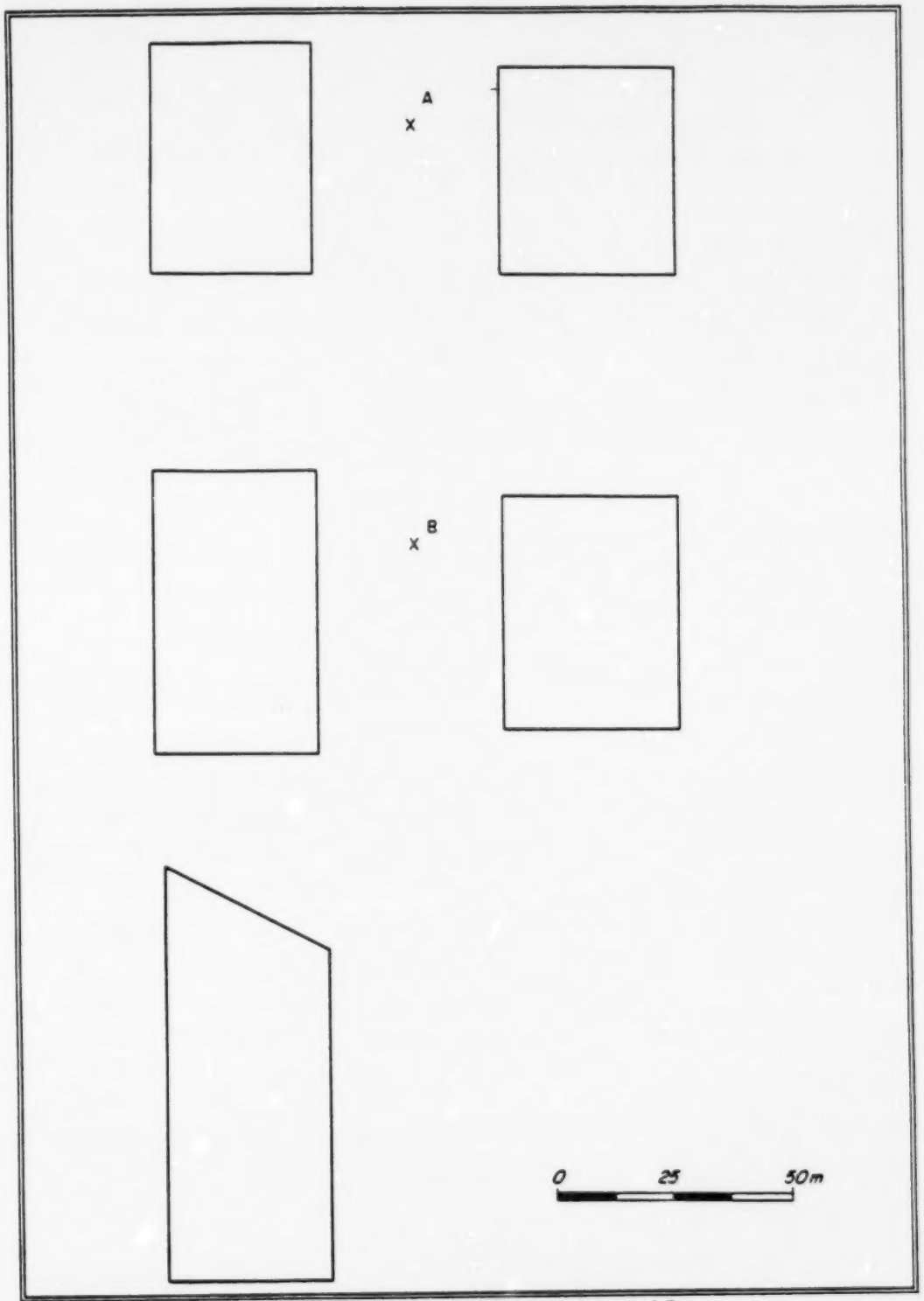


Figure 4.2.3: Five Excavation Problem - Location of Benchmarks A and B

		Principle Stresses A			Principle Stresses B		
Program	Run	Min	Int	Max	Min	Int	Max
MBEM	1	-2.6	25.9	43.0	-1.1	27.6	43.8
MBEM	2	-2.9	26.3	43.7	-0.7	28.0	44.1
MBEM	3	-2.1	26.8	44.2	0.45	28.2	44.1
MBEM	4	-0.5	26.2	44.8	0.28	27.7	44.3
BEAP	3	-2.3	26.6	44.2	-0.07	28.1	44.2
BEAP	4	-2.4	26.8	44.6	-0.4	28.3	44.4
Primitive (pre-mining)		16.9	24.2	36.3	17.3	24.9	36.6

TABLE 4.2.2 PRINCIPAL STRESSES AT A AND B FOR LOAD CASE 2

Agreement between principal stress directions for the above runs was good for all runs, and the agreement between the intermediate and maximum principal stresses is also good for the three programs THREE, MBEM and BEAP. The major differences between the various programs occur with the minimum principal stress The following paragraph attempts to explain these differences.

When an excavation is created, the surfaces of the excavation will generally tend to move into the excavation. This is shown by the horizontal displacements of the section BC of the previous example (Figure 3.2.2.2). From this same example, it is seen that the displacements vary very rapidly at the corner B of the tunnel. In particular, the continuous quadratic variation elements are seen to be too stiff in that they do not move enough at the point C where maximum movement is expected to occur. Thus, when one face of an excavation is modelled with one quadratic element, the centre point of the face will behave too stiffly. If that same excavation face is modelled in three dimensions, with 8 node quadratic elements, then the problem is even worse, because there is no node at the centre of the face. By contrast, the constant displacement/traction elements of program MBEM tend to move too freely. Now the minimum principal stress at A and B is dependent largely upon the extent to which the nearby excavation faces move away from the points A and B, ie the more movement away from A and B, the smaller (or more tensile) is the minimum principal stress at A and B.

With program THREE, insufficient movement takes place and the minimum principal stress is too compressive. With program MBEM, too much movement takes place and the minimum principal stress is too tensile. The correct value, it is felt, lies close to that predicted by BEAP, using non-conforming quadratic elements.

In mining applications, the difference between the maximum and minimum principal stress is often as important as the maximum principal stress. It is often required to predict accurately zones of tension. For cavities such as were used in this example, it is felt that the non-conforming elements provide more efficient and accurate alternative to the more conventional conforming elements. A comparison of computer CPU times is given in the Table below:

Program	Run	Elements	Equations	CPU Time	Disk MB	Computer
THREE 110/82	1	46	468	0:30:18	1.75	UNIVAC
THREE	2	136	1269	1:24:57	12.8	110/82
THREE	3	207	2643	7:05:40	?	110/82
MBEM	1	414	1242	0:19:12	2.43	DG MV 8000
MBEM	2	736	2208	0:47:16	8.67	DG MV 8000
MBEM	3	953	2859	0:58:07	11.8	DG MV 8000
MBEM	4	2304	6912	1:50:25	21.3	DG MV 8000
MBEM	5	570				
BEAP	1	46	1242	1:27:54	6.2	DG MV 8000
BEAP	2	184	4968	3:31:00	23.5	DG MV 8000

TABLE 4.2.3 : COMPARISON OF EXECUTION TIMES FOR VARIOUS RUNS

NOTE:

1 UNIVAC 1100/82 is rated as a 4.2 MIP Machine
DG MV 8000 is rated as a 1.2 MIP Machine

2 These runs were conducted in 1986.

4.3 EIGHT EXCAVATION PROBLEM -- BEAP

This is another example, which is included here mainly to demonstrate how a large problem can be efficiently modelled by making use of a single element for many of the excavation faces. The mesh, together with four separate mining steps which were modelled for comparative purposes are shown in figures 4.3.1 and 4.3.2.

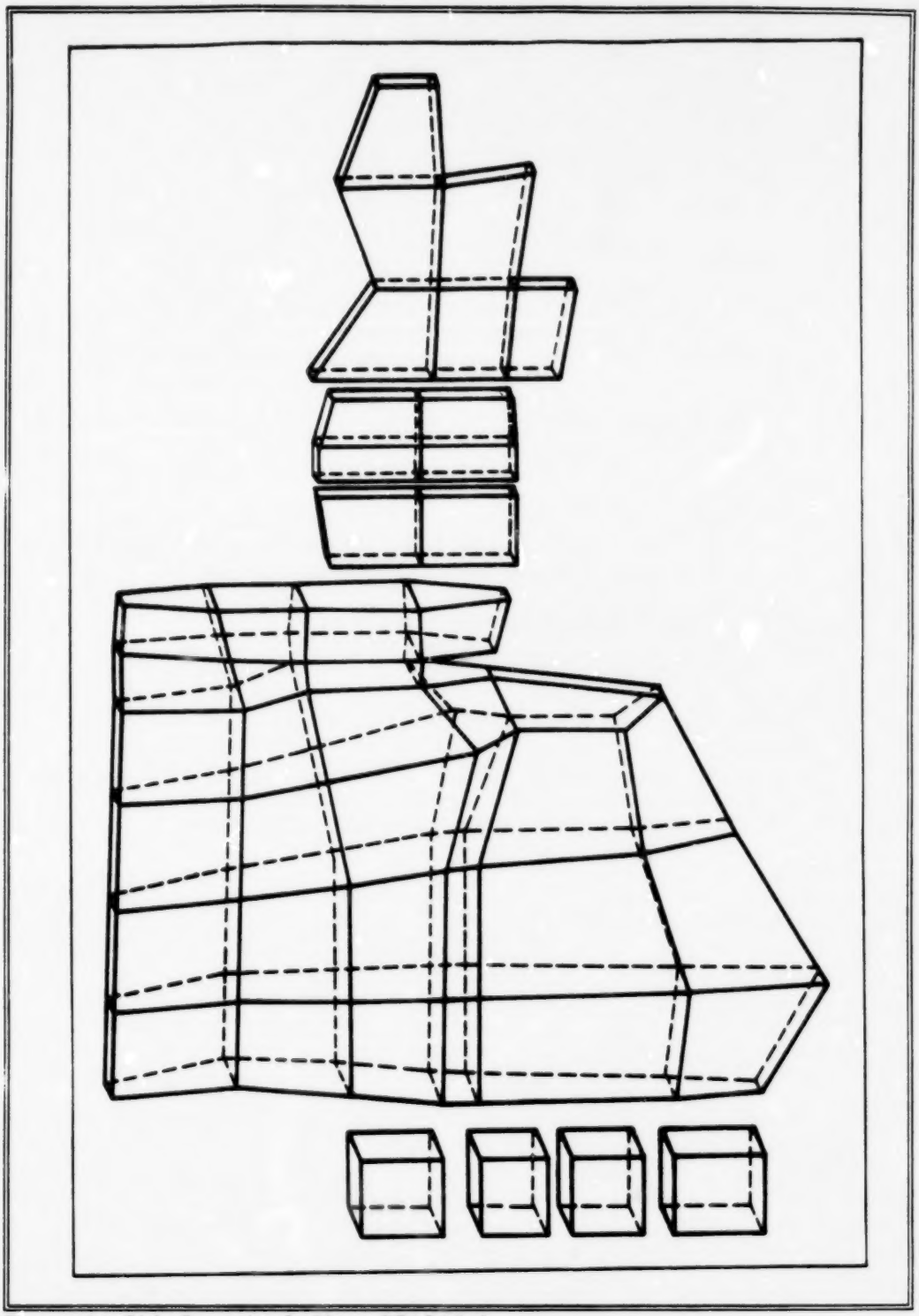


Figure 4.3.1: Eight Excavation Problem - BEAP Mesh

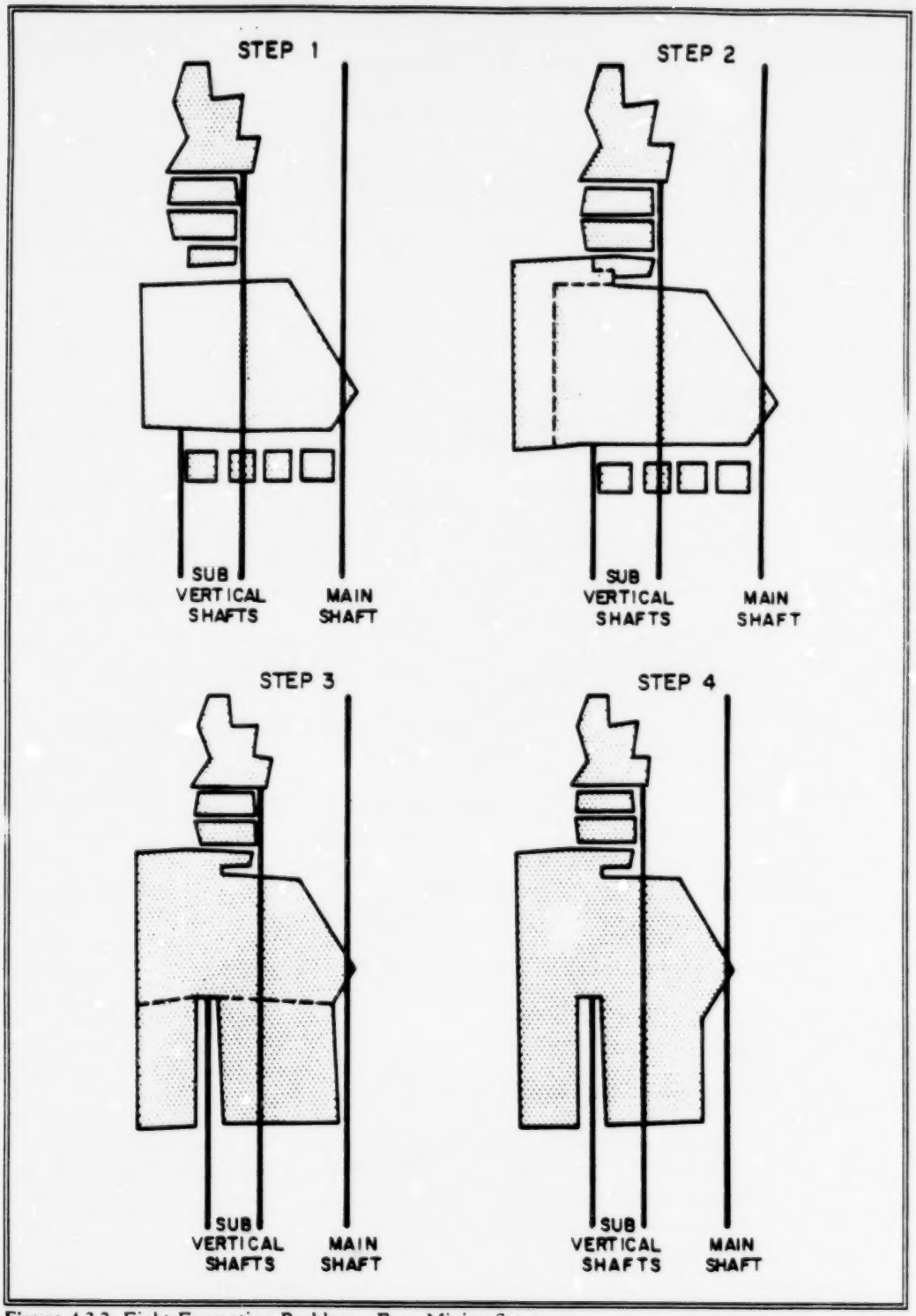


Figure 4.3.2: Eight Excavation Problem - Four Mining Steps

4.4 SHALLOW REEF EXAMPLE -- BEAP-M

This is essentially the same problem as was reported in section 3.3.3. However, the earth's surface is now modelled using 100 boundary elements. This enables a comparison between BEAP-M and BEAP-DD. Although this may seem an inconclusive test, it is, in fact, fairly rigorous. The movement of the earth's surface, and the surface/seam interaction are calculated using DD-DD type coefficients by BEAP-DD. These coefficients are based upon stress influence coefficients (Kd). On the other hand, with BEAP-M, the effect of the earth's surface on the underground is calculated using BE-DD type coefficients (Su-Dt) and the reverse effect using DD-BE type coefficients (Td). These latter coefficients are based upon displacement type coefficients

The results from the two comparisons are in good agreement. Examples for closures and rides for the BEAP-M runs are shown in figures 4.4.1 and 4.4.2. It is interesting to note that the general closure and ride profiles are smoother for the BEAP-M run than for the BEAP-DD runs (see figures 3.3.3.4 and 3.3.3.5), indicating that the earth's surface is being better modelled by the boundary elements than by displacement discontinuity elements as would be expected.

This problem also demonstrates the use of an incomplete subregion. Only the earth's surface is modelled, and that only a short distance from the displacement discontinuity grid. When the influence coefficients are calculated for the boundary elements numerous warnings are printed (M503) indicating the incomplete subregion. However, as the problem is still well formulated, these warnings may be disregarded in this case.

With the post processor facilities of BEAP-M, it is also possible to obtain various other representations of the model and results. These are shown in figures 4.4.3 to 4.4.5.

4.5 OPEN PIT/UNDERGROUND EXAMPLE

This example has been included here, although as yet it has not been solved on an IBM-PC. Geometry for the problem is shown in figure 1.1. This is the same problem which was modelled using the older MBEM program in 1983!. Data files have been prepared enabling figure 1.1 to be produced. However, disk space requirements exceed 30 Mbytes. Once BEAP-M has been installed on an IBM-PC compatible with 70 Mbytes or more of disk space, then this example can be run and should, hopefully, yield similar but more accurate results as compared with MBEM.

It is worth commenting on the boundary element mesh used here. Numerous triangular elements are used, which were convenient for the MBEM program. However, use of triangular elements within BEAP and BEAP-M is inefficient. A better mesh would have used fewer elements with fewer triangular elements as well. It can also be seen from figure 4.5.1 that the boundary element mesh has been set up to provide an element by element correspondence with the displacement discontinuity elements close to the underground pillars.

Close inspection of this mesh also shows a small gap between the boundary elements (open pit) and displacement discontinuity elements (underground), representing the final bench (15m) of the open pit which still requires to be mined in this simulation.

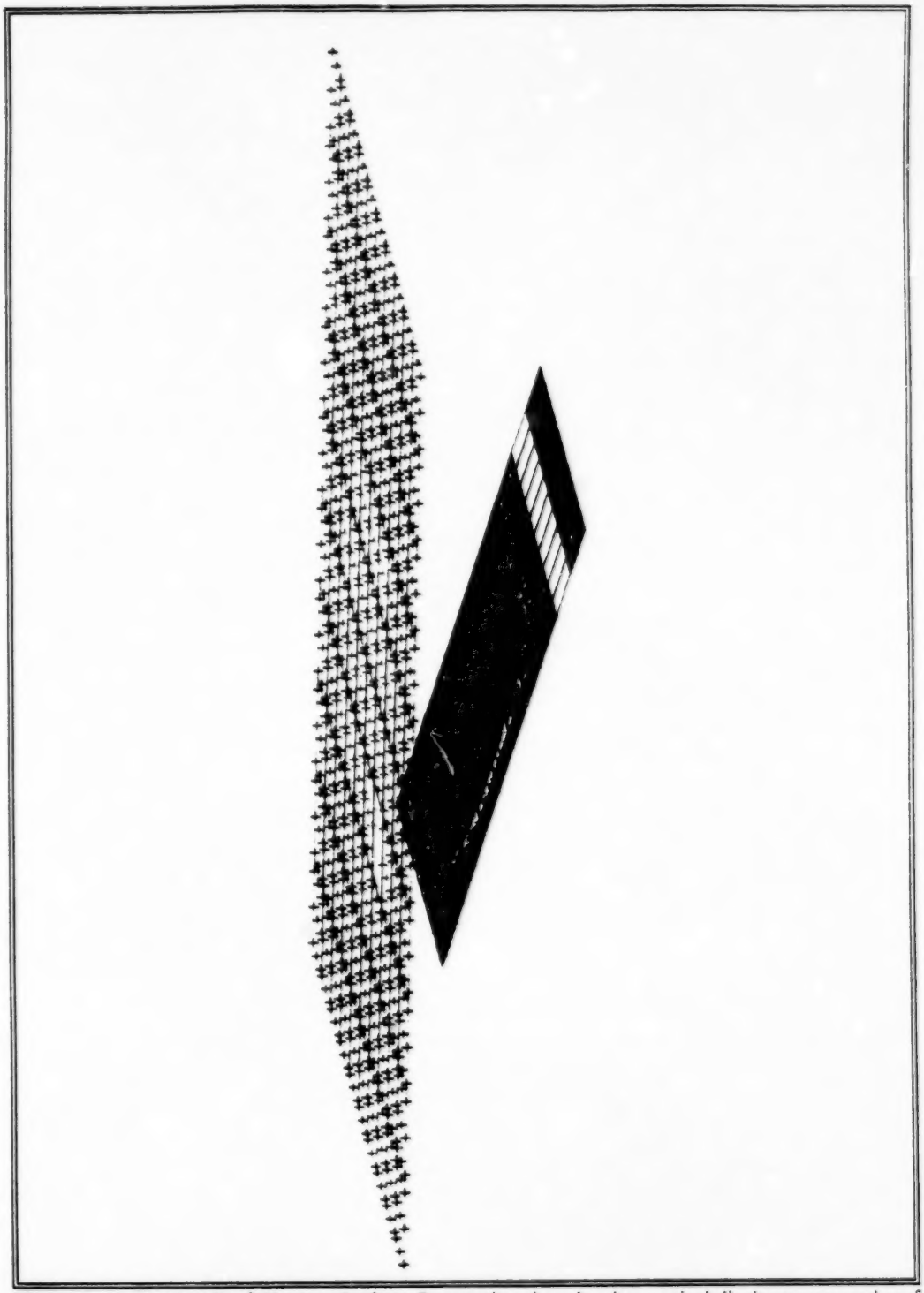


Figure 4.4.1: Dipping Reef Close to Surface, Perspective view showing vertical displacements and reef closures

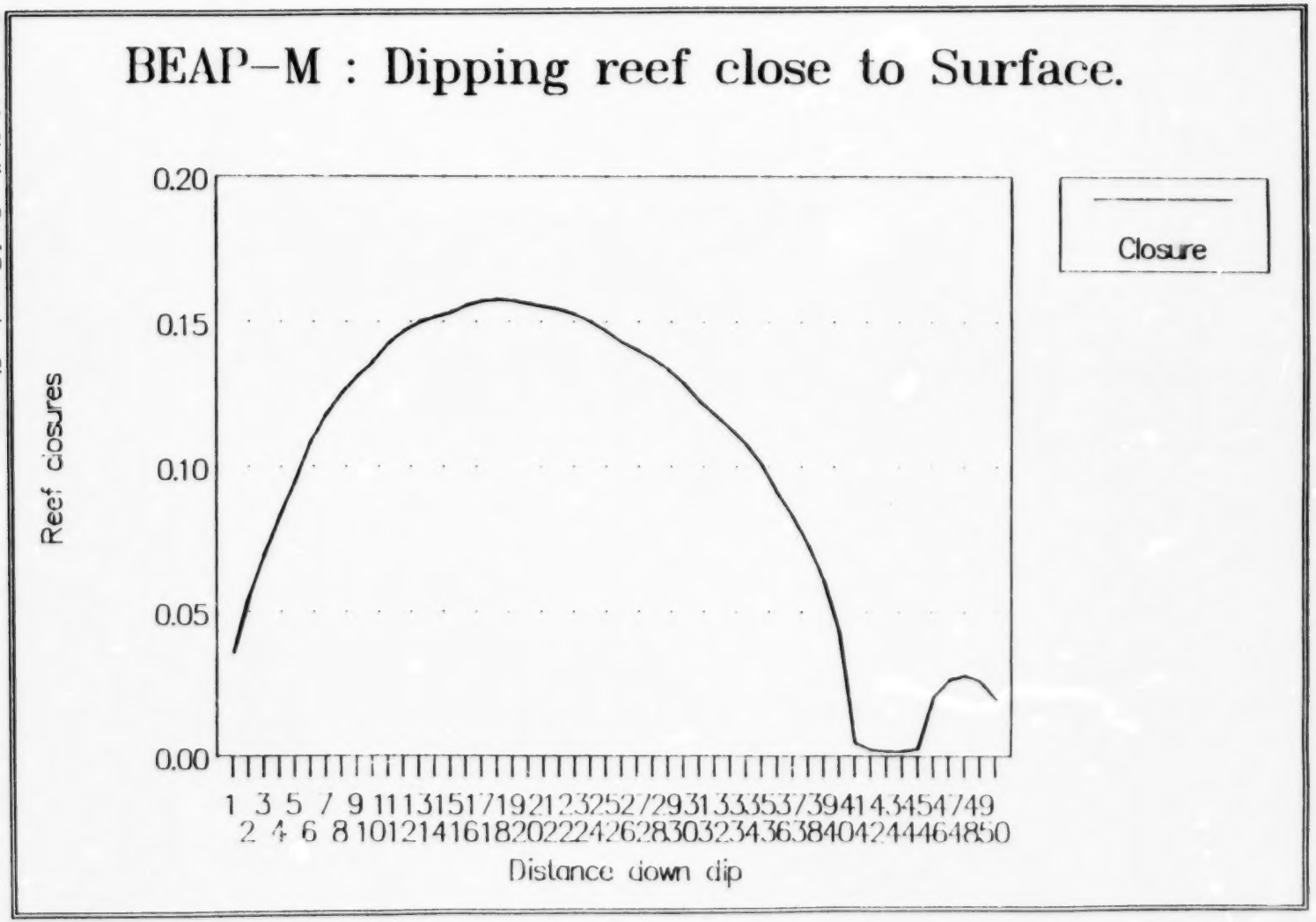


Figure 4.4.2: Shallow Reef Example - Closures

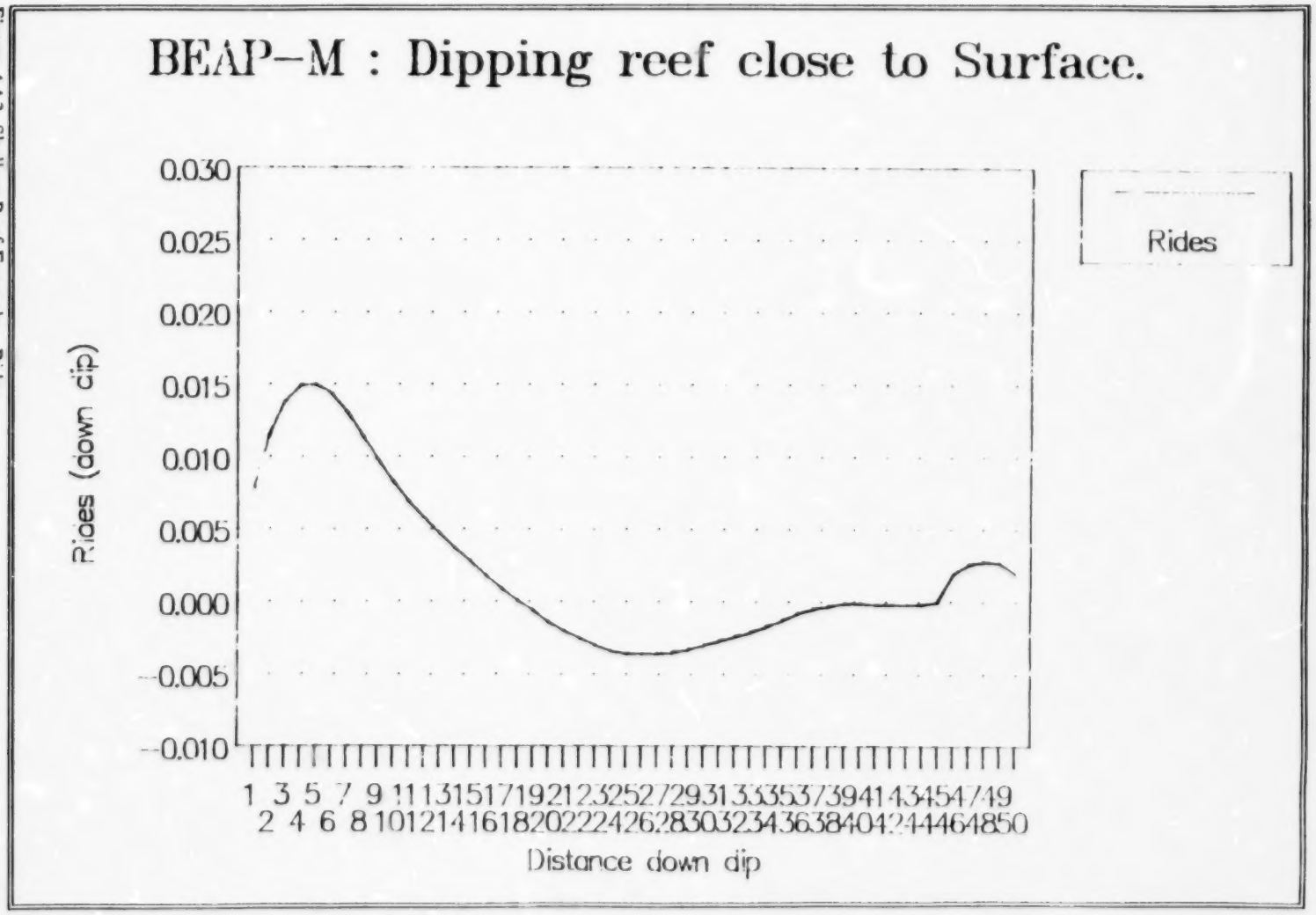


Figure 4.4.3: Shallow Reef Example - Rides

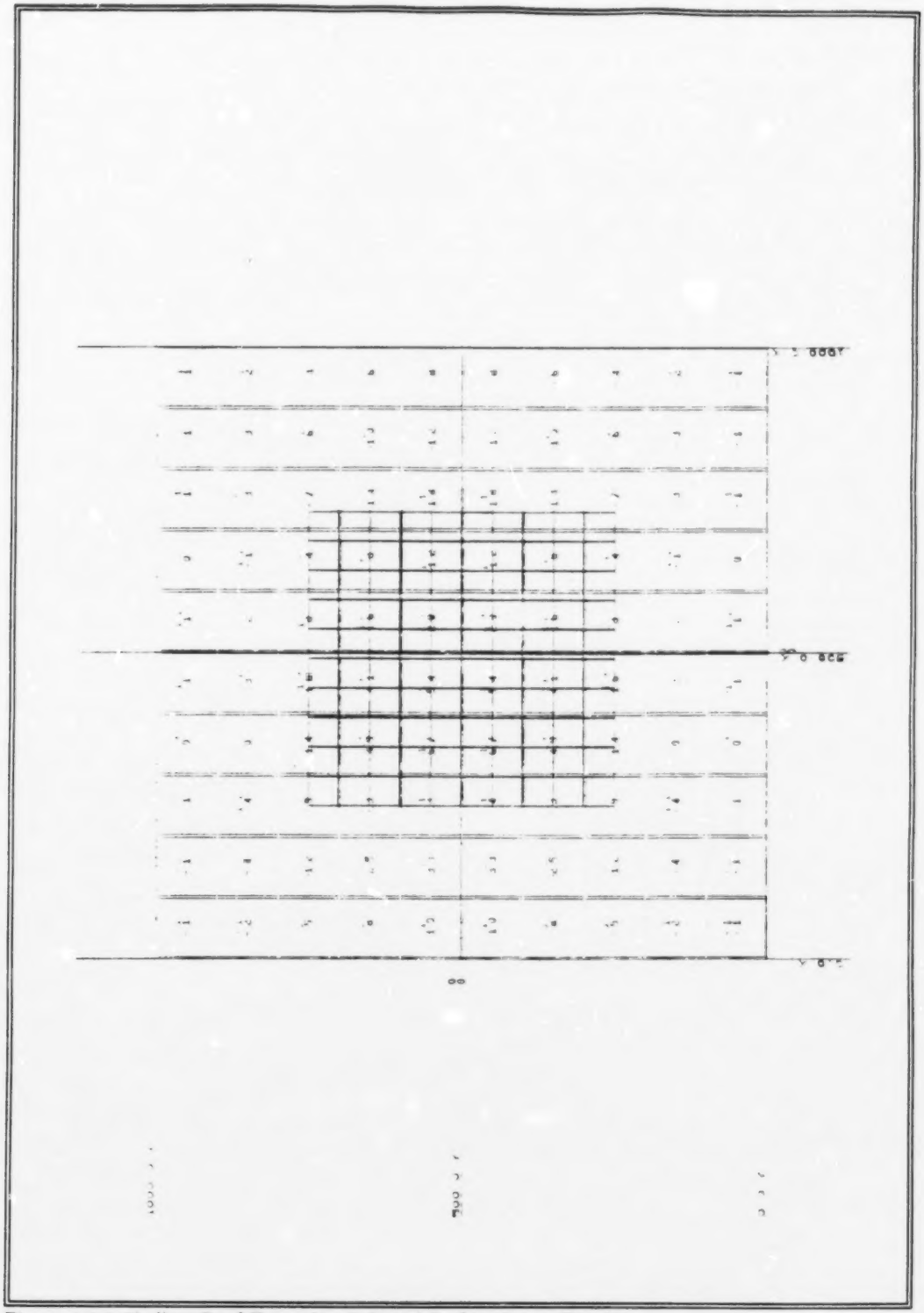


Figure 4.4.4: Shallow Reef Example - Selected Surface Stress Results

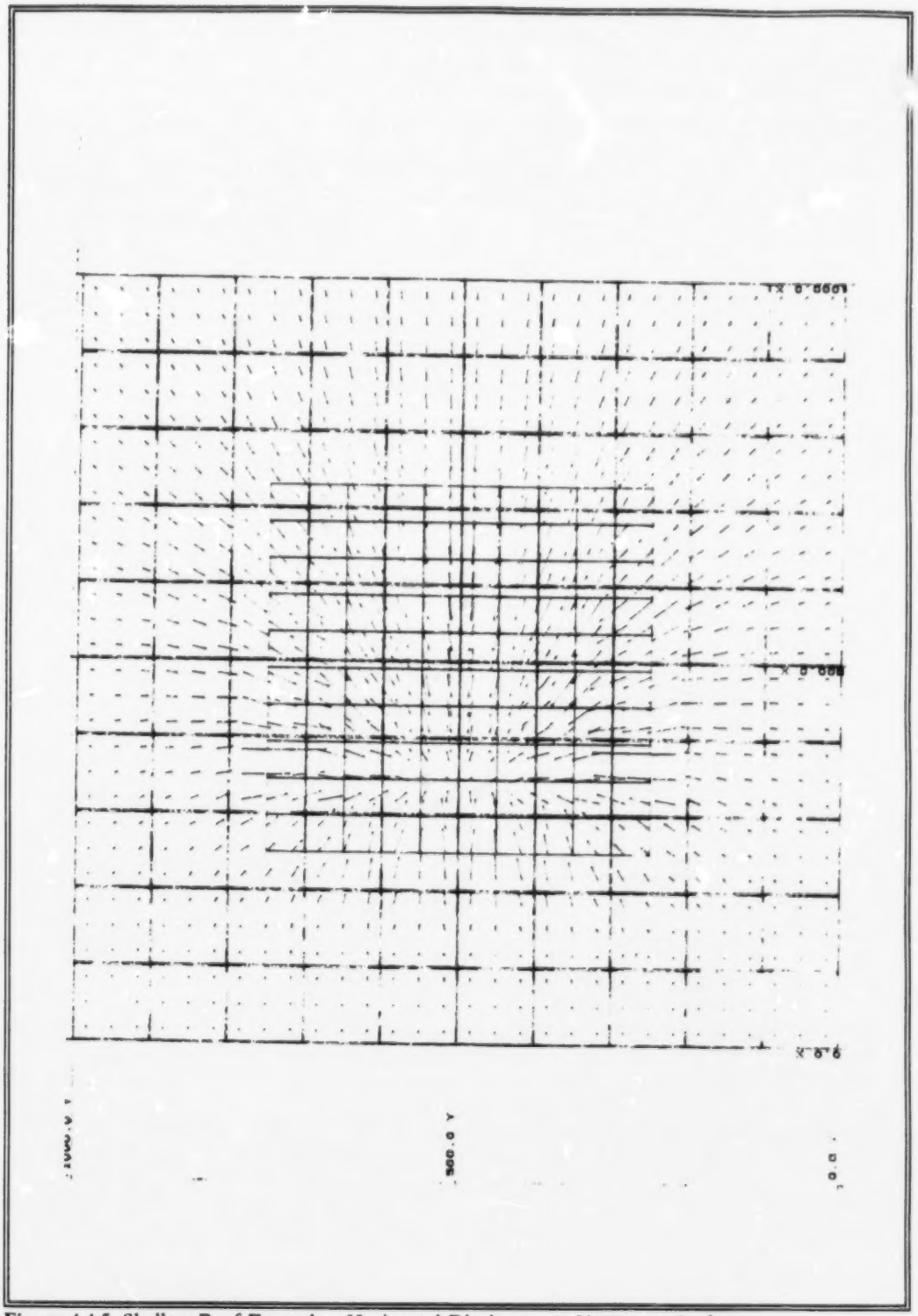


Figure 4.4.5: Shallow Reef Example - Horizontal Displacement Vectors at Surface

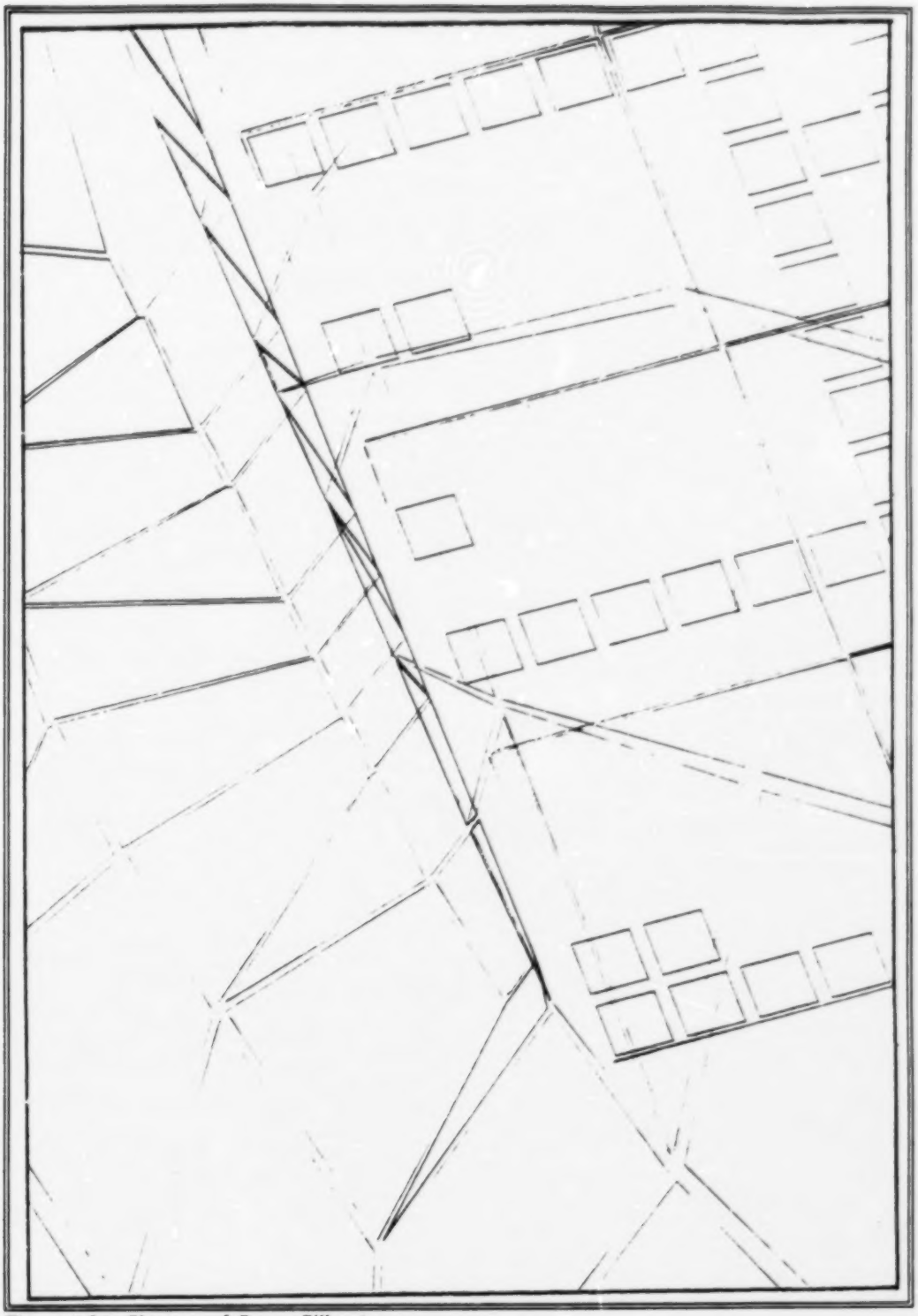


Figure 4.5.1: Close up of Crown Pillar Area

5.0 DISCUSSION

5.1 ORIGINAL OBJECTIVES

The extent to which the original design objectives have now been completed is shown on the following tables for the four programs. On average, it is felt that the numerical modelling as a whole has progressed well and that four useful and working packages have been developed.

Item	Description	Target/Objective	Where*
4a	Mining sequence	Change element codes step by step	Q D M
4b	Structural discontinuities faults and joints	Joint (finite) elements, and fault displacement discontinuity elements	Q D M
4c	Surface/underground interaction	Surface is easily modelled with relatively few boundary elements	(Q) B M
4d	Multiple or folded seams	Use displacement discontinuity elements	D M
4e	Improved windowing procedures	Windowing available through use of "frozen elements"	B D M
4f	Post failure material	Mohr Coulomb with dilatancy and non-linear Backfill seam elements	Q D M
4g	Failure criteria for post processing	Tension, Hoek's, Mohr, Extension based on current BEMPOST program	All
4h	Energy release	For multi step runs	D M
4i	User friendly/restart capability	Interactive or batch mode and modular	All
5	Fortran 77 Convert	Microsoft Fortran V3.3 (+ other Fortran also)	All
5	User Documentation	Separate manuals for each program	All
6	Pre-processing	Mesh pre-view (isometric) and geometry checks also digitizing in QUAD and BEAP-DD	All Q D
6	Post processing	Plotting of stress values etc., Plotting of benchmark data	All
-	Other	Calculate surface stresses, up to 4 or 5 sub-regions for BEAP	Various

* Program codes : B = BEAP, D = BEAP-DD, M = BEAP-M and Q = QUAD.

5.1.1 Mining sequence

Multi step capability has been built into the QUAD program. It is really only necessary here for multi-step runs in which non-linear behavior is being modelled. For linear runs, each 'mining' step could be modelled independently of others.

For the BEAP program, multi step runs can only be achieved by changing the geometry of the boundary element mesh. This has been done on various occasions (see figure 4.3.2).

Multi-step mining simulation is fully implemented into BEAP and BEAP-M. The following features ensure that multi step simulations may be effectively handled.

- Benchmarks for different steps may be stored or retrieved under program control
- The mining step number may be incremented or decremented under program control.
- Results from displacement discontinuity elements may be stored under program control
- Using DOS, relevant files from each step may also be stored, for example in subdirectories named STEP1, STEP2 etc.
- Influence coefficients may be calculated before the first mining step, so that when the layout of the displacement discontinuity elements changes, only the iterative solution part needs to be run to update the closures, rides and tractions.

5.1.2 Structural discontinuities (faults and joints)

Distributed joint elements are implemented in QUAD. These have been described above.

In BEAP-DD and hence BEAP-M a displacement discontinuity element may be used to model a fault element. Because of the 'folded seam' capability, this means that the intersection between a fault and underground mining could be modelled. However, as with multiple subregion problems (BEAP), the modelling of a discreet fault will substantially increase the running time (and disk space requirements) for any run.

5.1.3 Surface/ underground interaction

With the QUAD program, such interaction requires no special programming. The surface is modelled simply by terminating the finite element mesh there and not specifying any special boundary conditions at the surface nodes.

Surface/ underground interactions may be modelled with all the BEAP programs and the surface may be modelled using displacement discontinuity elements or boundary element, although use of the latter is more logical and more accurate.

5.1.4 Multiple or folded seams

Multiple or folded seams are only relevant to modelling with displacement discontinuity elements. With BEAP-DD, a lump is a group of 5 X 5 elements. Each lump can in principle have a unique sizing and orientation. In this way, as many folds as computing power permits may be modelled.

5.1.5 Improved windowing procedures

Windowing capability is achieved in one of two ways. Either elements may be 'frozen', in which case their tractions and displacements are fixed at their last calculated values or windowing may be achieved with BEAP-DD by using very different sized displacement discontinuity elements close to an area of interest.

In most cases, there does not appear to be any great need for windowing capability, but there will be some occasions when windowing will be useful.

With the boundary elements, a frozen element is activated simply as another boundary condition, similar to a free surface or mined element

5.1.6 Post failure material

Post failure material is intended to enable a re-distribution of stresses to occur if material strength is exceeded. The fault and joint capability described above falls into this category. In addition, QUAD allows the use of Mohr Coulomb failure criterion with or without dilation. No-tension material is also available. Backfill and total closure elements are also available within QUAD although these have not been used very often

The BEAP series of programs are essentially linear in their basic formulation. However, some degree of non-linear behavior is achieved in the backfill, total closure and fault elements of BEAP-DD.

5.1.7 Failure criteria for post processing

Separate post processing programs are used for QUAD and for the BEAP programs. However, each permits testing of calculated stresses against a Hoek's or Mohr Coulomb failure criterion. An extension strain criterion has also been used successfully in these programs.

5.1.8 Energy Release

Energy release rates may be calculated using the displacement discontinuity elements of BEAP-DD and BEAP-M. However, it is worth noting that the popularity of the ERR concept has greatly reduced over the past few years. In the South African gold mines, where the concept was most effectively used, there seems to be a growing demand for the use of the 'excess shear stress' concept (ESS). This is much closer in concept to the post processing failure criteria described in 5.1.7 above

5.1.9 User friendly/restart

The programs are, it is felt, user friendly and may readily be re-started at numerous points in the overall calculation sequence. However, especially as regards the three-dimensional programs, experience has shown that users need previous numerical modelling experience before these programs will be effectively used. Alternatively, a user must be trained up by some other experienced user before any serious modelling is attempted.

5.1.10 Fortran 77

The programs were originally developed on a Data General Computer using their Fortran 5. An earlier version of the QUAD program was implemented on the IBM main frame computer at Thompson, Manitoba prior to the commencement of this project. Subsequently, all source code has been converted to Microsoft Fortran V3.3. In the world of PC's, Microsoft compilers, it is believed, are generally accepted as standard. The Microsoft Fortran contains various features which are non ANSII standard. However, their use in these programs has been kept to a minimum. If the programs are to be transferred to other hardware, then conversion problems would result more from differences in the host operating system than from difficulties in compiling and linking the source code.

5.1.11 User documentation

User documentation has been prepared for each program. It MUST be emphasized here that the purpose of this documentation is NOT to teach users about boundary element or finite element analysis in general. If a user is not yet familiar with basic concepts and terms such as nodes, elements, boundary conditions and primitive stresses, then that user is not yet ready to run these programs.

The user documentation is intended to describe the format and purpose of the various input files. It describes how to obtain various printouts, but does not explain how these results should be interpreted.

A sound knowledge of operating system on which the programs are being run is advantageous, but not essential once the programs have installed.

The user documentation also provides various example data files, from which new data files for new projects can be developed.

5.1.12 Pre-processing

Pre-processing is available by means of graphical preview of a mesh geometry. In addition, polygons and polylines, used for delineation of mining outlines, may be displayed and checked before a model is run. Numerous checks are carried out by the programs on the geometry of the elements to assist with the input data checking.

With the BEAP programs, there is a convenient module (D301) which enables the automatic generation of many displacement discontinuity lumps form one single lump. For example if an array of 8 X 12 lumps is to be used, then only the four corner nodes for the overall grid need to be specified. Using M301, the other nodes and lumps may be generated.

Various digitizer facilities have also been implemented to assist with data preparation. With the QUAD program, nodes, elements and boundary conditions may be digitized. BEAP-DD and BEAP-M contain digitizer routines for digitizing mining layouts as polygons.

5.1.13 Post processing

The various post failure criteria have already been described. It is worth noting here, however, that all results are stored in disk files as they are being calculated. This means that various post-processing runs may be carried out. As the use of plotters and graphics screens usually requires other (non-FORTRAN) routines, the actual plotter and screen drivers have not been included as part of these

programs. However, plot files with simple and constant formats are prepared as part of the post processing capability. These files are then processed by other utility programs such as spreadsheet programs of GEMCOM's QUICKPLOT utility.

An attempt was made to get some graphics capability directly into these programs using (the CANMET recommended) GKS standards. However, this was unsuccessful. For example, the GKS package which was licensed was found to be incompatible with the Fortran compiler then in use. It was also found to be very slow for the plotting of large finite element meshes. It was concluded that the time would be much better spent in preparing interfaces to readily available commercial graphics packages and in developing other aspects of the programs.

5.2 OTHER FEATURES

There was no requirement in the original design objectives for this project for any boundary element modelling as a separate package. However, the development of the BEAP program is probably the most useful single product of task 4. BEAP uses the non-conforming quadratic boundary elements which are also new as regards mining applications.

6.0 CONCLUSIONS

It is concluded that the overall objectives as specified for the numerical modelling component of Task 4 have been met. With the wisdom of hind-sight, it is evident that certain of the objectives were slightly ambitious. Other objectives are now of less interest, particularly in light of the PC revolution.

Certain shortfalls should be mentioned. Firstly, in the five year period of the project, key personnel have moved between departments, companies and even countries. This has resulted in various communication problems. For example, the linking between the 'geomechanical database' and 'numerical models' has not been as well developed as was originally hoped. The movement of personnel has also made the training of users of the programs much more difficult.

Changing hardware conditions have also resulted in various inefficiencies in programming. For example, the BEAP and BEAP-DD programs were developed in Data General hardware. This computer had insufficient core memory to enable any real development of the BEAP-M program to take place. It was only after the programs had been loaded onto PC equipment that development of the BEAP-M software was possible. However, these programs are extremely CPU intensive. As such, mainframe computers may well be a better choice than a PC. However, such computers were not readily available as the general software development computer.

As regards the programs themselves, the list of objectives can often be interpreted in different ways. Certain objectives apply only to one or other of the packages. This absence of an absolute design target has, however, allowed the development of another useful modelling tool in its own right, namely the BEAP package.

On a more positive note, the end of this project has produced a substantial volume of useful software. Four programs made up of more than twenty thousand lines of source code have been developed. Most of the software has been well tested, although with the almost limitless combinations of options available, there are still certain bugs in the programs. These bugs are generally of a cosmetic nature, and should not effect the actual answers produced.

The BEAP program has already been successfully used on a number of projects and also installed on various hardware. Results from projects have been published in various journals. It is probably the first fully three-dimensional (industrial strength) boundary element package to be designed and implemented on a PC under the DOS operating system anywhere worldwide.

The MBEM program was the only package of its kind (using mixed boundary element and displacement discontinuity elements) five years ago. Now this functionality has been enhanced and replaced by BEAP-M. However, there still do not appear to be ANY comparable products available to the rock mechanics practitioner at the time of preparing this report.

One objective was a transfer of technology into Manitoba (and Canada in general). It is felt that the project has been very successful in this regard, although there is still a need for education of suitable users of this technology.

On balance, it is concluded that the project has been successfully completed, that the products developed represent useful additions to the current technology of numerical modelling in the field of rock mechanics.

7.0 REFERENCES

1 Deist, F H and Georgiadis, E, A computer system for three-dimensional elastic analysis using a boundary element approach. Chamber of Mines of South Africa, Research Report No 43/76, Project No. GS1510 (1976).

2 Hocking, G, Development and Application of Boundary Integral and Rigid Block Methods for Geotechnics. PhD Thesis, University of London, UK (1977).

3 Watson, J O and Cowling, R, Application of three-dimensional boundary element method to modelling of large mining excavations at depth. 5th Int Conf on Num Meth in Geomechanics, Nagoya (1985).

4 Diering, J A C, Further Developments of the Boundary Element Method with Applications in Mining. MSc Thesis, University of the Witwatersrand (1982).

5 Lachat, J C, A further Development of the Boundary Integral Technique for Elastostatics. PhD Thesis, University of Southampton, UK (1975).

6 Lachat, J C and Watson, J O, Effective Numerical Treatment of Boundary Integral Equations : A formulation for three-dimensional elastostatics. Int Jnl for Num Meth in Eng, 10, (1976) 991 - 1005.

7 Bannerjee, P K and Butterfield, R, Boundary Element Methods in Engineering Science. McGraw Hill (1982).

8 Watson, J O, Hermitian Cubic boundary elements for plane problems of fracture mechanics. Int Jnl of Struct Mech and Materials Sc, Ed J Gittus (1982).

9 Diering, J A C, Advanced elastic analysis of complex mine excavations using three-dimensional boundary element techniques. PhD Thesis Pretoria University, South Africa (1987).

10 Danson, D J, BEASY Boundary Element Analysis System, Computational Mechanics Centre, Manual : Southhampton, UK (1982).

11 Cruse, T A, Numerical solutions in three-dimensional elastostatics. Int Jnl Solids and Structures, 5, (1969) 1259 - 1274.

12 Zienkiewicz, O C, The Finite Element Method, London, McGraw Hill (1977).

13 Telles, J C F, A Self-adaptive Co-ordinate Transformation for Efficient Numerical Evaluation of General Boundary Element Integrals, Int Jnl for Num Meth in Eng, 24, (1987) 959 - 973.

14 Diering, J A C, Simulation of mining in non-homogeneous ground using the displacement discontinuity method. J S Afr Inst Min Metall, 80, No 7, (1980) 225 - 228.

15 Crouch, S L and Starfield, A M. Boundary Element Methods in Solid Mechanics. London, G Allen and Unwin (1983).

16 Brebbia, C A. The Boundary Element Method for Engineers. Pentech Press (1980).

17 Diering, J A C and Stacey, T R. Three-dimensional stress analysis : A practical planning tool for mining problems. Proc 20th Int Symp APCOM, Johannesburg, SAIMM, 1, (1987) 33 - 42.

18 Hoek, E and Brown, E T, Underground excavations in rock, IMM, London (1982).

19 Stacey, T. R., Van Heerden, W. L. and Diering, J. A. C.: "Analysis of excavations in jointed rock". Proc Conf on finite Element Methods in South Africa, Pretoria, South Africa. (1981)

20 Diering, J A C and Yu, Y S. Three-dimensional stress analysis : A formulation for mining problems. Computers and Geotechnics - Special issue on Mining Vol 5 No 2 Elsevier Appl Sc Publ (1988)